

46. ROČNÍK
MATEMATICKEJ
OLYMPIÁDY
NA STREDNÝCH ŠKOLÁCH
2. diel

Správa o riešení úloh zo súťaže konanej v školskom roku 1996/1997

38. MEDZINÁRODNÁ MATEMATICKÁ OLYMPIÁDA
9. MEDZINÁRODNÁ OLYMPIÁDA V INFORMATIKE

IUVENTA

S pomocou spolupracovníkov spracovali
RNDr. Karel Horák, CSc.,
Richard Kollár, Jana Višňovská,
Tomáš Vinař, Bronislava Brejová a členovia Úlohovej komisie MO.

© Richard Kollár za kolektív, 1998

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}\mathcal{E}\mathcal{X}$

ISBN 80-88893-19-4

Zadania súťažných úloh

KATEGÓRIA P

P – I – 1

Napište a odlaďte program na riešenie hry „15“. V tejto hre je v štvorcovej tabuľke 4×4 uložených 15 štvorcových kamienkov označených číslami $1, 2, \dots, 15$, jedno políčko je prázdne. *Povoleným krokom* je presunutie kamienka na susedné prázdne políčko. Povolený krok je jednoznačne určený poradovým číslom políčka, na ktorom stál presúvaný kamienok pred krokom. Políčka majú poradové čísla 1 až 16, pričom číslujeme po riadkoch od ľavého horného rohu. V znázornenom príklade sa zo situácie *A* dostaneme do situácie *B* krokom 6.

7	12	9	3
14	15		10
1	8	13	5
11	6	2	4

situácia *A*

7	12	9	3
14		15	10
1	8	13	5
11	6	2	4

situácia *B*

Úloha. Na začiatku hry sú kamienky rozložené náhodne. Účelom hry je povolenými krokmi dosiahnuť *koncovú situáciu K*, v ktorej sú kamienky usporiadané vzostupne podľa čísel a prázdne políčko je v pravom dolnom rohu:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Vstup. Váš program by mal

- zo súboru „P-I-1.INP“ načítať počiatočnú situáciu ako postupnosť čísel kamienkov oddelených medzerami (pričom napríklad situáciu *A* popisuje vstupná postupnosť „7 12 9 3 14 15 0 10 1 8 13 5 11 6 2 4“, kde znak 0 označuje prázdne políčko).

Výstup.

- ak koncová situácia je povolenými krokmi nedosiahnuteľná, do súboru „P-I-1.OUT“ vypísať správu „Nema riesenie“;
- ak koncová situácia je povolenými krokmi dosiahnuteľná:
 - do súboru „P-I-1.OUT“ vypísať (akúkoľvek) postupnosť povolených krokov, ktoré vedú ku koncovej situácii *K*;

- znázorniť postupnosť krokov vedúcu ku koncovej situácii pomocou semigrafiky alebo v grafickom režime. Toto znázornenie by malo byť interaktívne krokovateľné, napríklad po každom stlačení medzerníka by sa mal na obrazovke vykonať jeden krok postupnosti.

P – I – 2

Hovoríme, že množina bodov M v rovine je *schodovitá*, ak pre každú dvojicu (x_1, y_1) , (x_2, y_2) jej bodov platí

$$x_1 \neq x_2, \quad \text{pričom ak } x_1 < x_2, \quad \text{potom } y_1 > y_2.$$

To znamená, že ak body množiny M usporiadame podľa prvej súradnice, dostaneme „klesajúcu postupnosť druhých súradníc“.

Hovoríme, že dve neprázdne disjunktné množiny bodov A , B v rovine sú *separovateľné* (oddeliteľné), ak existuje priamka, ktorá rozdelí rovinu na dve polroviny tak, že sa v každej polrovine nachádzajú len body jednej z týchto množín, pričom separujúca priamka môže obsahovať body nanajvýš z jednej z nich.

Úloha. Sú zadané dve neprázdne disjunktné schodovité množiny bodov A a B . Vytvorte (čo najefektívnejší) algoritmus a napíšte program, ktorý zistí, či množiny bodov A , B sú separovateľné, a ak sú separovateľné, nájde separujúcu priamku.

Vstup. Vstupný súbor „P-I-2.INP“ obsahuje riadok s dvoma celými kladnými číslami m a n určujúcimi po rade počty prvkov množín A a B . Potom nasleduje m riadkov, z ktorých každý obsahuje dve reálne čísla – súradnice bodov množiny A a n riadkov rovnakého formátu obsahujúcich súradnice bodov množiny B . Zoznamy bodov A aj B sú usporiadané vzostupne podľa hodnôt prvej súradnice. Môžete predpokladať, že vstup je zadaný korektne.

Výstup. Výstupom programu (zápis do súboru „P-I-2.OUT“) je riadok obsahujúci tri reálne čísla a , b a c oddelené medzerami, ktoré sú koeficientmi všeobecnej rovnice niektorej separujúcej priamky $ax + by + c = 0$. Ak separujúca priamka neexistuje, výstupom je trojica čísel „0 0 0“.

P – I – 3

Orient-expres prechádza n ($n < 100$) rôznymi štátmi (očísľujeme ich od 1 po n). Každý štát má svoju vlastnú menu (tiež ich očísľujeme: štát k nech má menu k) a tieto meny sú medzi sebou voľne zameniteľné. V štáte k môžeme nakupovať iné meny, ale len za menu k . Cestujeme zo štátu i do štátu j ($i < j$), potrebujeme teda nakúpiť menu j . Kladné reálne číslo $K_{(i,j)}$ určuje, koľko jednotiek meny j dostaneme pri priamej výmene za jednu jednotku meny i . Niekedy sú ale výhodnejšie nepriame obchody: zameniť menu i za l a potom menu l za j ($i < l < j$), príp. i za l , l za m a m za j ($i < l < m < j$) a podobne.

Predpokladajme napríklad, že cestujeme zo štátu 1 do 3 a meny štátov 1, 2 a 3 sú vony, tugriky a dongy. Ak za 1 von dostaneme 1,50 tugrikov alebo 1,25 dongov a za 1

tugrik dostaneme 0,90 dongov, je výhodnejšie zameniť vony najprv za tugriky a potom za dongy (za sto vonov takto dostaneme 135 dongov namiesto 125 dongov, ktoré možno získať priamou výmenou).

Úloha. Napíšte a odlaďte program, ktorý po načítaní tabuľky výmenných kurzov K dokáže odpovedať na otázky typu „aký je najvýhodnejší (priamy alebo nepriamy) výmenný kurz pri ceste zo štátu i do štátu j ?“.

Vstup. Vstupom programu (v súbore „P-I-3.INP“) je

- kladné celé číslo n ($n < 100$) – počet štátov,
- $\frac{n(n-1)}{2}$ kladných reálnych čísel – výmenné kurzy:

$$\begin{array}{ccccccc} K_{(1,2)} & K_{(1,3)} & K_{(1,4)} & \dots & K_{(1,n)} & & \\ & K_{(2,3)} & K_{(2,4)} & \dots & K_{(2,n)} & & \\ & & & \ddots & \vdots & & \\ & & & & & & K_{(n-1,n)} \end{array}$$

- kladné celé číslo p – počet vstupných dvojíc („otázok“), ktoré budú nasledovať,
- p dvojíc celých čísel i, j ($1 \leq i < j \leq n$) – počiatočné a koncové štáty ciest.

Výstup. Výstupom programu (súbor „P-I-3.OUT“) bude p kladných reálnych čísel: najvýhodnejšie výmenné kurzy pre jednotlivé cesty. Reálne čísla stačí vypisovať s presnosťou na dve desatinné miesta.

Príklad vstupu a zodpovedajúceho výstupu:

P-I-3.INP: 4 1.50 1.25 2.20 0.90 1.30 1.55 3 1 2 1 3 1 4

P-I-3.OUT: 1.50 1.35 2.20

P – I – 4

Učebný text.

Kombinačný obvod pozostáva zo *vstupov obvodu* (označených I_1, I_2, \dots, I_n), z *hradiel* (označených H_1, H_2, \dots, H_k) a z *výstupov obvodu* (označených O_1, O_2, \dots, O_m). Každé hradlo má jeden alebo viac vstupov označených x_1, x_2, \dots, x_p a jeden výstup označený z . Každý vstup hradla H_i je napojený buď na niektorý vstup obvodu alebo na výstup niektorého iného hradla H_j , $j < i$. Každý výstup obvodu O_i je napojený na výstup niektorého hradla.

Vstupy a výstupy obvodu, ako aj vstupy a výstupy hradiel môžu byť v logických stavoch 0 (false) a 1 (true). Výpočet obvodu je taktovaný: ak vstupy hradla H sú v čase t v logických stavoch $x_1[t], x_2[t], \dots, x_p[t]$ výstup H bude v čase $t + 1$ v logickom stave $z[t + 1]$ určenom vstupmi a typom hradla. Ak je na tento výstup napojený napríklad vstup x' hradla H' , bude v čase $t + 1$ vstup x' v logickom stave $z[t + 1]$ – t.j. logický stav sa po spojnicách šíri bez zdržania.

Budeme používať tri typy hradiel: NAND, NOR (v úlohe P-I-4 s dvoma, v úlohách P-II-4 a P-III-3 s viacerými vstupmi) a NOT (s jedným vstupom). Závislosť výstupu

týchto troch typov hradiel na vstupoch popisujú nasledujúce tabuľky (uvádzame príklady hradiel NAND a NOR s dvoma a troma vstupmi):

$x_1[t]$	$x_2[t]$	$z[t+1]$
0	0	1
0	1	1
1	0	1
1	1	0

NAND

$x_1[t]$	$z[t+1]$
0	1
1	0

NOT

$x_1[t]$	$x_2[t]$	$z[t+1]$
0	0	1
0	1	0
1	0	0
1	1	0

NOR

$x_1[t]$	$x_2[t]$	$x_3[t]$	$z[t+1]$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

NAND

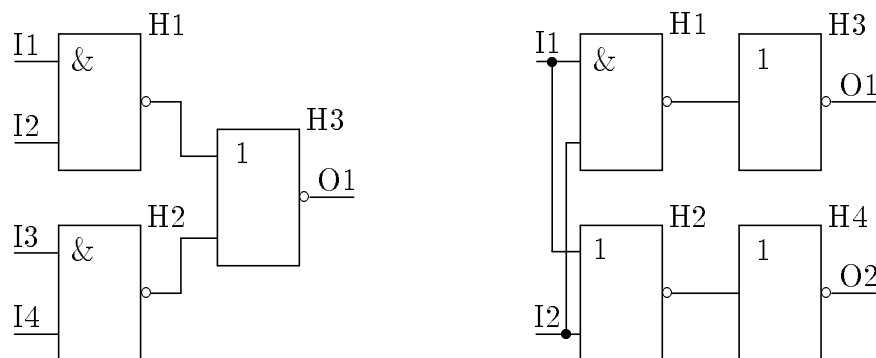
$x_1[t]$	$x_2[t]$	$x_3[t]$	$z[t+1]$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

NOR

To znamená, že hradlo typu NAND má výstupnú hodnotu 0 práve vtedy, ak všetky jeho vstupy majú hodnotu 1. Hradlo typu NOR má výstupnú hodnotu 1 práve vtedy, ak všetky jeho vstupy majú hodnotu 0.

Vstupom výpočtu je n -tica logických stavov vstupov obvodu I_1, I_2, \dots, I_n . Logický stav vstupov obvodu sa počas výpočtu nemení. Hradlá postupne reagujú na logické stavy svojich vstupov a menia stavy svojich výstupov. Samozrejme, ak sa v čase t ustália (t.j. ďalej sa nemenia) logické stavy vstupov hradla, od času $t+1$ sa ustáli aj jeho výstup. Po konečnom čase sa ustália aj logické stavy výstupov obvodu. Výsledkom výpočtu je m -tica logických stavov výstupov obvodu O_1, O_2, \dots, O_m .

V prípade obvodu na obrázku vľavo (hradlá $H1, H2$ typu NAND sú označené $\&$, hradlo $H3$ typu NOR je označené 1, logické stavy vstupov obvodu sú 0, 1, 1, 1) bude od času $t=1$ logický stav výstupu $H1$ rovný 1 a logický stav výstupu $H2$ rovný 0. Od času $t=2$ bude logický stav výstupu hradla $H3$ rovný 0, čo je aj výsledkom výpočtu.



V obvode vpravo vystupujú aj hradlá $H3$, $H4$ typu NOT (sú označené 1, rovnako ako NOR – majú ale len jeden vstup). Po dvoch taktoch (v čase $t = 2$) sa výstupy obvodu ustália a výsledkom bude „utriedený vstup“: na $O1$ bude menší a na $O2$ väčší z logických stavov vstupov obvodu. Chovanie tohto obvodu by sme mohli popísať tabuľkou:

I1	I2	O1	O2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Súťažné úlohy.

Vstup. Napíšte program, ktorý zo vstupného súboru načíta

- tri čísla – počet vstupov obvodu (n), počet hradiel (k) a počet výstupov obvodu (m),
- popis kombinačného obvodu vo vhodnom formáte, ktorý sami navrhnete,
- číslo p udávajúce počet vstupných n -tíc núl a jednotiek,
- p vstupných n -tíc núl a jednotiek – vstupy kombinačného obvodu.

Výstup. Do výstupného súboru potom vypíše

- p výstupných m -tíc núl a jednotiek – výsledky výpočtu obvodu so zadanými vstupnými n -ticami.

Úloha. Navrhňte kombinačný obvod s dvoma vstupmi ($I1 = x$, $I2 = y$) a dvoma výstupmi ($O1 = c$, $O2 = s$) tak, aby realizoval sčítanie dvoch jednobitových čísel: $2 \cdot c + s = x + y$. To znamená, že výstup s bude obsahovať paritu súčtu a výstup c bude obsahovať prenos (carry). Obvod nakreslite a popíšte aj vo vstupnom formáte pre váš program.

P – II – 1

Hrací plán pozostáva z $2n$ políčok umiestnených v jednom rade vedľa seba ($n \geq 3$). Na každom políčku sa nachádza modrý alebo červený kamienok, celkový počet modrých a červených kameňkov je rovnaký (n).

Cieľom hry je usporiadať kamienky: v ľavej polovici hracieho plánu majú byť červené, v pravej polovici modré kamienky. Povolenu operáciou v hre je obrátenie poradia troch alebo štyroch susedných kameňkov, t.j. zmena $abc \rightarrow cba$, resp. $abcd \rightarrow dcba$, kde abc ($abcd$) sú susedné kamienky na hracom pláne. Operáciu môžeme jednoznačne určiť pomocou dvoch čísel: pozície p najľavejšieho políčka meneného úseku a dĺžky l meneného úseku ($1 \leq p \leq 2n - l + 1$, $l = 3, 4$). Riešením hry je postupnosť dvojíc (p, l) , ktorá počiatočnú situáciu prevedie na usporiadanú.

Vstup. Napíšte program, ktorý načíta zo vstupného súboru P-II-1.INP

- číslo n ($3 \leq n \leq 1000$) – počet modrých resp. červených kameňkov (v prvom riadku súboru);
- počiatočnú situáciu – postupnosť $2n$ znakov m (reprezentuje modrý kamienok) a c (červený kamienok) v druhom riadku súboru.

Výstup. Do výstupného súboru P-II-1.0UT potom vypíše postupnosť dvojíc čísel $p \ l$ (krokov riešenia v tvare popísanom vyššie; každú dvojicu na samostatný riadok) ukončenú riadkom obsahujúcim dve nuly. Vypísaná postupnosť predstavuje riešenie zadanej hry.

Poznámky: Uvedomte si, že každá počiatočná situácia je riešiteľná. Váš program nemusí nájsť optimálne riešenie (tzn. dĺžka vygenerovanej postupnosti nemusí byť najmenšia možná), snažte sa však nájsť efektívny algoritmus.

P – II – 2

Nech A je postupnosť bodov $A[1], A[2], \dots, A[n]$ ($n \geq 1$), kde bod $A[i]$ má celočíselné súradnice $(A[i].x, A[i].y)$. Takáto postupnosť sa nazýva *schodovitá*, ak pre každé i menšie ako n platí:

$$\begin{aligned} A[i].x < A[i+1].x, & \quad A[i+1].x - A[i].x \leq 2, \\ A[i+1].y < A[i].y, & \quad A[i].y - A[i+1].y \leq 2 \end{aligned}$$

a vzdialenosť medzi bodmi $A[i+1]$ a $A[i]$ je menšia ako $2\sqrt{2}$.

Robot sa môže pohybovať po ceste určenej bodmi schodovitej postupnosti, pričom táto cesta začína v bode $A[1]$, pokračuje cez body $A[2], \dots, A[n-1]$ a končí v bode $A[n]$, pričom medzi každými dvoma bodmi postupnosti sa robot pohybuje po úsečke.

Z cesty robota možno odstrániť úsečku $A[i-1]A[i]$ tak, že body $A[i]$ až $A[n]$ posunieme tak, aby sa bod $A[i]$ dostal do bodu $A[i-1]$ (všetky body posúvame o rovnakú vzdialenosť v x -ovom aj y -ovom smere). Bod $A[i-1]$ potom vylúčime z postupnosti. Všimnite si, že vzniknutá postupnosť je takisto schodovitá, pričom počet bodov sa zmenšil o jedna.

Úloha. Dané sú dve schodovité postupnosti A ($|A| = n$) a B ($|B| = m$). Robot R_A sa pohybuje po ceste určenej postupnosťou A a robot R_B po ceste určenej postupnosťou B .

Nájdite čo najefektívnejší algoritmus, ktorý určí najmenšie počty úsečiek, ktoré je potrebné odstrániť z ciest robotov R_A, R_B tak, aby sa oba roboty (za predpokladu, že sa zo svojich počiatočných bodov začnú pohybovať naraz rovnakou rýchlosťou) pohybovali súbežne (t.j. ich vzájomná poloha je stále rovnaká) a skončili v tom istom čase v poslednom bode príslušne upravených schodovitých postupností.

P – II – 3

Orient-expres prechádza n ($n < 100$) rôznymi štátmi. Budeme používať označenie a pravidlá pre nákup mien v jednotlivých štátoch zavedené v úlohe P-I-3 (*pozri na strane 2*).

Niektoré zo štátov sa rozhodli podporovať turizmus (samozrejme nie svojich vlastných občanov), a preto cestujúceho, ktorý vystúpi z Orient-expresu, aby si vymenil peniaze, na niekoľko dní zdržia. Nezáporné celé číslo $T[i]$ určuje, koľko dní musí stráviť turista v štáte i , ak si tam vymenil peniaze (náklady na pobyt v ďalšom neuvažujeme

a pre jednoduchosť budeme predpokladať, že cestujúci na medzistanicích vystupujú len za účelom výmeny peňazí). Po uplynutí $T[i]$ dní pokračuje cestujúci v ceste (Orient-expres jazdí denne). Žiadny cestujúci nie je ochotný obetovať viac než určitý počet dní (nanajvýš 10) na „turistický pobyt“ na medzistanicích.

Predpokladajme napríklad, že cestujeme zo štátu 1 do 3 a meny štátov 1, 2 a 3 sú vony, tugriky a dongy. Ak za 1 von dostaneme 1,50 tugrikov alebo 1,25 dongov a za 1 tugrik dostaneme 0,90 dongov, je výhodnejšie zameniť vony najprv za tugriky a potom za dongy (za sto vonov takto dostaneme 135 dongov namiesto 125 dongov získateľných priamou výmenou). Ak ovšem $T[2]$ je 5 a môžeme si dovoliť zdržanie nanajvýš tri dni, druhá možnosť neprichádza do úvahy (všimnite si, že $T[1]$ a $T[3]$ nehra v tejto úvahe žiadnu úlohu).

Úloha. Napíšte program, ktorý po načítaní tabuľky výmenných kurzov K a tabuľky zdržaní T dokáže odpovedať na otázky typu „aký je najvýhodnejší (priamy alebo nepriamy) výmenný kurz pri ceste zo štátu i do štátu j , ak sme ochotní stráviť nanajvýš k dní turistikou?“.

Vstup. Vstupom programu (v súbore P-II-3.INP) je

- kladné celé číslo n ($n < 100$) – počet štátov;
- $\frac{n(n-1)}{2}$ kladných reálnych čísel – výmenné kurzy

$$\begin{array}{cccccc} K_{(1,2)} & K_{(1,3)} & K_{(1,4)} & \dots & K_{(1,n)} \\ & K_{(2,3)} & K_{(2,4)} & \dots & K_{(2,n)} \\ & & & \ddots & \vdots \\ & & & & K_{(n-1,n)} \end{array}$$

- n nezáporných celých čísel – zdržania $T[1], T[2], \dots, T[n]$;
- kladné celé číslo p – počet vstupných trojíc („otázok“), ktoré budú nasledovať;
- p trojíc celých čísel i, j, k ($1 \leq i < j \leq n, k \leq 10$) – počiatočné a koncové štáty ciest a maximálne zdržania.

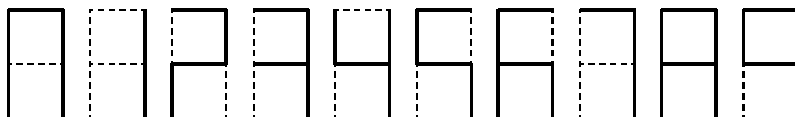
Výstup. Výstupom programu (v súbore P-II-3.OUT) bude p kladných reálnych čísel: najvýhodnejšie výmenné kurzy pre jednotlivé cesty. Reálne čísla stačí vypisovať s presnosťou na dve miesta za desatinnou bodkou.

Poznámka: Pri riešení nemusíte brať ohľad na obmedzenie veľkostí polí dané niektorými prekladačmi programovacích jazykov.

P – II – 4

Zadanie tejto úlohy vychádza zo študijného textu zaradeného pred úlohou P-I-4, pozri na strane 3. Hradlá typu NAND a NOR môžu mať ale tentokrát (narozdiel od úlohy P-I-4) viac než dva vstupy – bližšie pozri spomínaný študijný text.

Úloha. Vašou úlohou je navrhnúť a popísať kombinačný obvod so 4 vstupmi a 7 výstupmi na ovládanie LED-displeja pre znázornenie čísiel 0 až 9:



Vstupy obvodu označíme x_0, x_1, x_2, x_3 a výstupy $y_0, y_1, y_2, y_3, y_4, y_5, y_6$. Vstupy obvodu budeme interpretovať ako binárny zápis čísiel 0 až 9 (x_3 je najvýznamnejší bit):

x_3	x_2	x_1	x_0	význam
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
\vdots	\vdots	\vdots	\vdots	\vdots
1	0	0	1	9

Binárne zápisy čísiel väčších ako 9 nás nebudú zaujímať a výstupy obvodu pre tieto vstupy môžu byť ľubovoľné.

Výstupy obvodu zodpovedajú segmentom displeja, kde

- y_0 ovláda horný vodorovný segment,
- y_1 ovláda ľavý horný zvislý segment,
- y_2 ovláda pravý horný zvislý segment,
- y_3 ovláda stredný vodorovný segment,
- y_4 ovláda ľavý dolný zvislý segment,
- y_5 ovláda pravý dolný zvislý segment,
- y_6 ovláda dolný vodorovný segment.

Presné chovanie obvodu môžeme popísať tabuľkou, kde 0 znamená „nesvieti“, 1 „svieti“ a ? je „ľubovoľná hodnota“ (0 alebo 1).

$x_3x_2x_1x_0$	y_0	y_1	y_2	y_3	y_4	y_5	y_6
0000	1	1	1	0	1	1	1
0001	0	0	1	0	0	1	0
0010	1	0	1	1	1	0	1
0011	1	0	1	1	0	1	1
0100	0	1	1	1	0	1	0
0101	1	1	0	1	0	1	1
0110	1	1	0	1	1	1	1
0111	1	0	1	0	0	1	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1
iné	?	?	?	?	?	?	?

Obvod môžete popísať ľubovoľným spôsobom, ktorý jednoznačne určí hradlá obvodu: ich typ (vstup x_i , NAND, NOR, NOT, výstup y_i) a napojenie ich vstupov (napríklad obrázok).

P – III – 1

Sú dané súradnice n modrých a n červených bodov v rovine. Žiadne tri z týchto bodov neležia na jednej priamke a všetky sú navzájom rôzne.

Úloha. Napíšte čo najefektívnejší algoritmus, ktorý nájde n úsečiek takých, že každá z nich má jeden koncový bod v červenom bode a druhý v modrom bode a žiadne dve úsečky nemajú ani jeden spoločný bod. V prípade, že riešenie neexistuje, algoritmus o tom vypíše vhodnú správu.

P – III – 2

Nech A je postupnosť bodov $A[1], A[2], \dots, A[n]$ ($n \geq 1$), kde bod $A[i]$ má celočíselné súradnice $(A[i].x, A[i].y)$. Takáto postupnosť sa nazýva *schodovitá*, ak pre každé i menšie ako n platí:

$$\begin{aligned} A[i].x < A[i+1].x, & \quad A[i+1].x - A[i].x \leq 2, \\ A[i+1].y < A[i].y, & \quad A[i].y - A[i+1].y \leq 2 \end{aligned}$$

a vzdialenosť medzi bodmi $A[i+1]$ a $A[i]$ je menšia ako $2\sqrt{2}$.

Daný je počiatkový bod S s celočíselnými súradnicami $(S.x, S.y)$ a koncový bod K so súradnicami $(K.x, K.y)$.

Úloha. Napíšte čo najefektívnejší algoritmus, ktorý určí schodovitú postupnosť s najmenším možným počtom bodov takú, že jej prvým bodom bude S a posledným bodom K . (Ak teda postupnosť bude mať n bodov, potom $A[1] = S$ a $A[n] = K$.) V prípade, že takáto postupnosť neexistuje, algoritmus o tom vypíše vhodnú správu. Dôležitou súčasťou riešenia je **dôkaz**, že množina, ktorú nájde váš algoritmus, má skutočne najmenší možný počet bodov.

P – III – 3

Zadanie tejto úlohy vychádza zo študijného textu zaradeného pred úlohou P-I-4, strana 3. Hradlá typu NAND a NOR môžu mať ale tentokrát (narozdiel od úlohy P-I-4) viac než dva vstupy – bližšie pozri spomínaný študijný text.

Úloha. Vašou úlohou je navrhnúť a popísať kombinačné obvody s 8 vstupmi x_1, x_2, \dots, x_8 a jedným výstupom pre nasledujúce funkcie:

- **parita** – výstupná hodnota je 1 práve vtedy, ak medzi vstupnými hodnotami je párny počet jedničiek,
- **majorita** – výstupná hodnota je 1 práve vtedy, ak medzi vstupnými hodnotami je aspoň 5 jedničiek,
- **aspoň2** – výstupná hodnota je 1 práve vtedy, ak medzi vstupnými hodnotami sú aspoň 2 jedničky.

P – III – 4

Hrací plán je pás políček očíslovaných zľava doprava číslami 0, 1, 2, atď. Na niektorých políčkach hracieho plánu je umiestnený kamienok, celkový počet kamienkov je od 1 po 4. *Poradové číslo* kamienka je počet kamienkov, ktoré ležia na pláne naľavo od neho. Prvý kamienok zľava má teda poradové číslo 0 a posledný $k - 1$. Každú situáciu hry možno popísať počtom kamienkov a číslami políček, na ktorých sú jednotlivé kamienky umiestnené. Môžete predpokladať, že žiaden kamienok nebude na políčku s číslom väčším ako $2^{31} - 1$. *Koncová situácia* je situácia, v ktorej sa kamienky nachádzajú na políčkach 0, 1, \dots , $k - 1$.

Hru hrajú dvaja hráči, ktorí striedavo ťahajú kamienkami. V jednom ťahu je povolené posunúť jeden z kamienkov o ľubovoľný počet políček pri dodržaní nasledujúcich podmienok:

- kamienky sa posúvajú doľava,
- na každom políčku je vždy nanajvýš jeden kamienok,
- počet kamienkov, ktoré sú naľavo od posúvaného kamienka sa posunutím nezmení.

V koncovej situácii teda nie je možné spraviť ďalší ťah a hra končí. Vyhráva hráč, ktorý urobil posledný krok. Môžete predpokladať, že na začiatku hra nie je v koncovej situácii.

Úloha. Napíšte program (ďalej označovaný ako *hráč*), ktorý bude hrať túto hru, a to vždy ako začínajúci hráč (prvý na ťahu). Druhého hráča bude zastupovať knižnica popísaná nižšie (ďalej označovaná ako *protivník*), ktorú použijete vo vašom programe. Vašou úlohou bude napísať program, ktorý čo najúspešnejšie hrá proti ľubovoľnému protihráčovi (t.j. proti každej knižnici spĺňajúcej uvedený popis).

Knižnice sú uložené v súboroch HRA_C.LIB (pre C/C++) prípadne HRA_P.TPU (pre PASCAL). Knižnica obsahuje niekoľko funkcií, resp. procedúr s nasledujúcimi hlavičkami:

```

/* C/C++ */
#ifdef __cplusplus
extern "C" {
#endif
#define MAXKAM 4
typedef long THraciPlan[MAXKAM];

void inicializuj(int* pocetKamienkov, THraciPlan hraciPlan);
void mojKrok(int kamienok, long dlzkaKroku);
void tvojKrok(int* kamienok, long* dlzkaKroku);
#ifdef __cplusplus
}
#endif

{Pascal}
interface
    const MAXKAM = 4;
    type THraciPlan = array[0..MAXKAM-1] of longint;

```

```

procedure inicializuj(var pocetKamienkov : integer;
                     var hraciPlan : THraciPlan);
procedure mojKrok(kamienok: integer; dlzkaKroku : longint);
procedure tvojKrok(var kamienok: integer;
                  var dlzkaKroku : longint);

```

Funkcia `inicializuj` inicializuje hru a vráti počiatočnú situáciu hry. V parametri `pocetKamienkov` vráti počet kamienkov na hracom pláne a v poli `hraciPlan` vráti pozície kamienkov, pričom `hraciPlan[i]` je pozícia kamienka s poradovým číslom i , pre $i = 0, 1, \dots, \text{pocetKamienkov} - 1$, hodnoty ostatných prvkov poľa sú nedefinované.

Pomocou funkcie `mojKrok` oznamujete protihráčovi svoj krok. V parametri `kamienok` odovzdávate poradové číslo posúvaného kamienka a v parametri `dlzkaKroku` počet políčok, o ktorý posúvate zvolený kamienok.

Pomocou funkcie `tvojKrok` získate naopak súperov ťah, pričom význam argumentov je taký istý ako vo funkcii `mojKrok`.

Váš program na začiatku svojho behu zavolá funkciu `inicializacia` a potom striedavo volá funkcie `mojKrok` a `tvojKrok`, pričom začína funkciou `mojKrok`. Môžete predpokladať, že protivník hrá korektne. Ak hráč spraví víťazný ťah, t.j. ťah, po ktorom nastane koncová situácia, funkcia `mojKrok` ukončí program. Ak naopak protivník svojim nasledujúcim ťahom zvíťazí, funkcia `tvojKrok` ukončí program.

Ak váš program nedodrží opísané poradie volania procedúr alebo zadá nepovolený krok ako argument funkcie `mojKrok`, váš program prehráva. Takisto je zakázané ukončiť program pred koncom hry.

Poznámky:

- Na skúšku máte k dispozícii jedného protihráča. Tento protihráč načítava počiatočnú pozíciu zo súboru `HRA.IN` (tento súbor obsahuje počet kamienkov a ich pozície).
- Váš program môže byť testovaný aj s inými protihráčmi.

P – III – 5

V starovekom meste Kuru archeológovia objavili zvláštny trezor. Pozostával zo štyroch dverí. Dvere sa museli otvoriť v správnom poradí, lebo inak celú miestnosť zaliala voda. Na každých dverách bolo vytesané nejaké zviera. Na stene archeológovia rozlúštili text, ktorého preklad do slovenčiny znie:

Ako prvé otvor dvere, na ktorých je buď lev alebo orol.

Ako druhé otvor dvere, na ktorých je buď had alebo škorpión.

Ako tretie otvor dvere, na ktorých je buď orol alebo had.

Ako druhé otvor dvere, na ktorých je buď lev alebo had.

Ak predpokladáme, že všetky tieto tvrdenia sú pravdivé, môžeme určiť poradie otvárania dverí. Z druhého a štvrtého tvrdenia vyplýva, že ako druhé treba otvoriť dvere, na ktorých je had. Z tretieho tvrdenia a z faktu, že had je na druhých dverách, vyplýva, že orol je na tretích dverách. Z prvého tvrdenia a z faktu, že orol je na tretích dverách vyplýva, že lev je na prvých dverách. Pre škorpióna ostali štvrté dvere.

Úloha. Napíšte čo najefektívnejší program, ktorý bude riešiť takýto typ hlavolamov. Je daný počet dverí n ($1 \leq n \leq 1000$). Na každých dverách je vytesané iné zviera, pričom tieto zvieratá označíme číslami od 1 po n . Ďalej je dané číslo m ($1 \leq m \leq 1000$) a m tvrdení, pričom každé tvrdenie môže byť jedného z dvoch tvarov:

- „Ako i -te treba otvoriť dvere, na ktorých je zviera j “, kde $1 \leq i, j \leq n$. Toto tvrdenie budeme kódovať trojicou čísel 1 i j.
- „Ako i -te treba otvoriť dvere, na ktorých je buď zviera j alebo zviera k “, kde $1 \leq i, j, k \leq n$, $j \neq k$. Toto tvrdenie budeme kódovať štvoricou čísel 2 i j k.

Riešenie hlavolamu je také poradie otvárania dverí, v ktorom sa každé dvere vyskytujú práve raz, a ktoré spĺňa všetky zadané tvrdenia.

Vstup. Vstupný súbor obsahuje niekoľko hlavolamov. Prvý riadok súboru obsahuje počet hlavolamov v súbore. Pre každý hlavolam vstupný súbor obsahuje blok údajov. V prvom riadku bloku sa nachádzajú čísla n a m . Každý z ďalších m riadkov obsahuje kód jedného tvrdenia. Jednotlivé bloky údajov sú oddelené vždy jedným prázdny riadkom.

Výstup. Pre každý hlavolam vypíšte do výstupného súboru jeden riadok, ktorý bude obsahovať:

- text „Neexistuje riesenie“, ak hlavolam nemá ani jedno riešenie,
- text „Viacero rieseni“, ak hlavolam má viac ako jedno riešenie,
- riešenie hlavolamu (t.j. čísla zvierat vytesaných na dverách v takom poradí ako treba dvere otvárať; za sebou idúce čísla oddelíte práve jednou medzerou), ak hlavolam má práve jedno riešenie.

Riešenia súťažných úloh

KATEGÓRIA P

P – I – 1

Najprv sa pokúsime odpovedať na otázku, kedy úloha nemá riešenie. Po získaní určitých praktických skúseností s hrou zistíme, že každú počiatočnú situáciu S dokážeme previesť na situáciu $K = 1, 2, \dots, 13, 14, 15, 0$ alebo $N = 1, 2, \dots, 13, 15, 14, 0$, kde 0 predstavuje prázdne políčko. Po ďalších skúsenostiach získame presvedčenie (podporené aj prvým príkladom v zadaní úlohy), že situácia N je neriešiteľná. To ale znamená tiež neriešiteľnosť počiatočnej situácie S . Inak by sme N vyriešili prevodom na S („spätným chodom“ – zopakovaním prevodu S na N v opačnom poradí) a následným prevodom S na K . Ak teda budeme mať algoritmus, ktorý ľubovoľnú počiatočnú situáciu S prevedie buď na K , alebo na N , budeme vedieť rozhodnúť o riešiteľnosti S – stačí sa pozrieť, či sme skončili v K alebo v N .

Riešiteľné a neriešiteľné situácie sa dokonca dajú matematicky charakterizovať. Táto charakterizácia nám poskytne dôkaz neriešiteľnosti N , a tým dá úvahám v predchádzajúcom odstavci pevný základ.

Zavedieme najprv pojem *počet inverzií danej situácie S* . Vypíšeme si čísla kamienkov S za sebou, ale vynecháme 0 (prázdne políčko). Pre situáciu A uvedenú v zadaní by sme dostali postupnosť 7, 12, 9, 3, 14, 15, 10, 1, 8, 13, 5, 11, 6, 2, 4. Inverzia je taká dvojica (i, j) členov tejto postupnosti, že i predchádza v postupnosti j , a pritom $i > j$. Napríklad 12 a 9 tvorí inverziu v postupnosti A , ktorá má spolu

$$6 + 10 + 7 + 2 + 9 + 9 + 6 + 0 + 4 + 5 + 2 + 3 + 2 + 0 + 0 = 65 \quad \text{inverzií.}$$

Ďalším číslom, ktoré nás bude zaujímať, je číslo riadku (1 až 4), v ktorom sa nachádza prázdne políčko (pre A je to 2). *Charakteristikou $C(A)$ situácie A* nazveme súčet týchto dvoch čísel (počtu inverzií a čísla riadku prázdneho políčka). Všimnite si, že $C(A) = 65 + 2 = 67$, $C(K) = 0 + 4 = 4$ a $C(N) = 1 + 4 = 5$. Čo sa stane s charakteristikou, keď prejdeme povoleným krokom zo situácie S do S' ? Rozlíšime dva prípady:

- krok vodorovným smerom (doľava alebo doprava): ani jeden zo sčítancov sa nezmení, teda $C(S) = C(S')$.
- krok zvislým smerom (hore alebo dole): číslo riadku sa zmení o 1. Počet inverzií sa zmení o 1 alebo o 3 (dokážte!).

Celkovo teda $C(S)$ patrí do rovnakej zvyškovej triedy po delení dvoma ako $C(S')$. Inými slovami povedané, parita charakteristiky sa povolenými krokmi nezmení. Ak má začiatková situácia nepárnu charakteristiku, určite sa nedá previesť na koncovú situáciu K , a preto je neriešiteľná. Neriešiteľná je teda aj situácia N , rovnako ako všetky situácie, ktoré sa na ňu dajú previesť postupnosťou povolených krokov.

Teraz uvedieme algoritmus, ktorý ľubovoľnú počiatočnú situáciu prevedie buď na K , alebo na N – presnejšie na neriešiteľnú situáciu

$$N' = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 15, 13, 14, 12, 0), \quad C(N') = 5 + 4 = 9.$$

Myšlienka algoritmu je jednoduchá: postupne dostaneme na svoje miesto kamienky 1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 10, 14, 11 a 0 (prázdne políčko na pozíciu 16). Pokiaľ je po skončení kamienok 15 na svojom mieste, je aj 12 v poriadku a máme K , inak sme dospeli k N' . Pre každý kamienok (napr. k), ktorý chceme takto dostať na miesto, definujeme *cyklus* – cyklickú postupnosť susedných pozícií obsahujúcu

- cieľovú pozíciu pre kamienok k
- pozíciu, na ktorej kamienok k práve leží
- prázdnu pozíciu.

Kamienky budeme presúvať po tomto cykle, dokiaľ sa kamienok nedostane na svoje miesto (toto ešte upresníme neskôr). Môžeme dokonca predpísať, kde má skončiť prázdne políčko. Budeme posúvať kamienky po cykle (určenom kamienkom K torý) tak, že sa K torý dostane na pozíciu K am a pozícia NP (Nechaj Prázdnu) ostane prázdna. Cykly zásadne obchádzajú pozície, ktorým už algoritmus pridelil správny kamienok. Jednotlivé cykly vyzerajú nasledovne (uvádzame ich v smere putovania prázdneho políčka):

Ktorý	Kam	NP	cyklus	dĺžka
1	1	2	1,2,3,4,8,12,16,15,11,7,6,10,14,13,9,5,1	16
2	2	3	2,3,7,8,12,16,15,11,10,14,13,9,5,6,2	14
3	3	8	3,4,8,12,16,15,11,10,14,13,9,5,6,7,3	14
4	8	12	5,6,7,8,12,16,15,11,10,14,13,9,5	12
5	5	6	5,6,7,8,12,16,15,11,10,14,13,9,5	12
6	6	7	6,7,11,12,16,15,14,13,9,10,6	10
7	7	12	7,8,12,16,15,14,13,9,10,11,7	10
8	12	16	9,10,11,12,16,15,14,13,9	8
9	9	14	9,13,14,15,16,12,11,10,9	8
13	14	15	10,11,12,16,15,14,10	6
10	10	15	10,14,15,16,12,11,10	6
14	15	16	11,15,16,12,11	4
11	11	16	11,12,16,15,11	4

Musíme ešte vysvetliť dva problematické body.

- Potom, čo sme dostali kamienok 1 na pozíciu 1 (a pozícia 2 je prázdna), môže sa stať, že kamienok 2 je na pozícii 4, teda mimo druhý cyklus. To vyriešime uplatnením krokov 3, 4, 8 (kamienok 2 sa ocitne na pozícii 3 a pozícia 8 ostane prázdna). Obdobný problém je aj s kamienkom 6.
- Druhý problém sa týka umiestenia „rohových“ kamienkov 4, 8, 13 a 14 – práve pre ne je K torý \neq K am v tabuľke. Keď už sme umiestnili kamienky 1, 2 a 3, žiadny cyklus

obchádzajúci pozície 1, 2 a 3 neprechádza pozíciou 4. Preto kamienok 4 neumiestňujeme na pozíciu 4 priamo (pokiaľ, samozrejme, ešte na nej neleží), ale pozíciu 8 a pozíciu 12 necháme prázdnu – pozri cyklus pre kamienok 4. Potom mechanicky uplatníme postupnosť jedenástich krokov získanú skúšaním, ktorá dostane 4 na svoje miesto, pričom dočasne premiestni aj kamienok 3. Podobne postupujeme v prípade 8, 13 a 14.

Program najprv vypočíta postupnosť krokov vedúcu na K (príp. N'), uloží ju do poľa a overí, či je počiatočná situácia riešiteľná. Ak áno, obnoví vopred uloženú počiatočnú situáciu a predvedie (krokováním po opakovanom stlačení klávesu) nájdené riešenie.

Na dokončenie popisu algoritmu ešte potrebujeme odhadnúť počet krokov riešenia, aby sme mohli nadefinovať pole na uloženie jednotlivých krokov nájdeného riešenia. Pozrime sa najprv na prevádzanie jedného cyklu dĺžky d . Rozmiestnenie kamienkov v tomto cykle sa behom prevádzania mení, ale všetkých možných rozmiestnení je $d(d - 1)$, lebo rozmiestnenie je určené polohou kamienka *Ktorý* a polohou prázdneho políčka. To znamená, že pri prevádzaní cyklu sa vykoná nanajvyš $d(d - 1)$ krokov. Z tabuľky teda určíme, že celkový počet krokov uložených do poľa je

$$16 \cdot 15 + 2 \cdot 14 \cdot 13 + 2 \cdot 12 \cdot 11 + 2 \cdot 10 \cdot 9 + 2 \cdot 8 \cdot 7 + 2 \cdot 6 \cdot 5 + 2 \cdot 4 \cdot 3 = 1\,254.$$

K tomuto číslu ešte musíme pripočítať korekciu pre kamienky 2 a 6 (spolu 6 krokov) a 4 ošetrenia rohových kamienkov po jedenástich krokoch (spolu 44 krokov), čo spolu dáva 1 304 krokov.

P – I – 2

Nech $A[1], A[2], \dots, A[m]$ sú body prvej množiny, $B[1], B[2], \dots, B[n]$ sú body druhej množiny, $m, n \geq 1$. Na súradnice bodov týchto množín sa budeme odvolávať pomocou bodkovej notácie, t.j. $A[k].x$ je x -ová súradnica a $A[k].y$ je y -ová súradnica bodu $A[k]$.

Pokiaľ $m = 1$ a zároveň $n = 1$, rozlíšime dva prípady:

- body $A[1]$ a $B[1]$ sú totožné, teda množiny nie sú separovateľné,
- body $A[1]$ a $B[1]$ sú rôzne, potom množiny sú separovateľné, napríklad priamku kolmú na úsečku $A[1]B[1]$ a prechádzajúcu stredom tejto úsečky.

Bez ujmy na všeobecnosti ďalej predpokladajme, že $m > 1$ (inak je možné množiny preznačiť). Základné myšlienky postupu riešenia zhrnieme v nasledujúcich krokoch:

- Vytvoríme konvexný obal množiny A . Vďaka špeciálnemu tvaru množiny je to jednoduchšie ako vo všeobecnom prípade. Táto časť algoritmu vyžaduje lineárny čas vzhľadom na počet bodov množiny A , t.j. $O(m)$.
- Zistíme polohu bodu $B[1]$ voči konvexnému obalu množiny A . Ak bod $B[1]$ leží v konvexnom obale množiny A , potom množiny A a B nie sú separovateľné. Pokiaľ bod $B[1]$ neleží vnútri konvexného obalu množiny A , potom už poznáme jeden bod množiny B , ktorý neleží vnútri. Zároveň určíme polohu bodu $B[1]$ vzhľadom k priamke určenej bodmi $A[1]$ a $A[m]$. Pokiaľ bod $B[1]$ leží v kladnej polrovine, potom v prípade, že množiny sú separovateľné, očakávame všetky body množiny B v tejto

polrovine, inak analogicky uvažujeme o zápornej polrovine. Táto časť algoritmu vyžaduje lineárny čas vzhľadom na počet bodov množiny A , t.j. $O(m)$.

- Tento a ďalšie kroky prevádzame len keď bod $B[1]$ neleží v konvexnom obale množiny A . Vytvoríme konvexný obal množiny B . Tento krok algoritmu vyžaduje lineárny čas vzhľadom k počtu bodov množiny B , t.j. $O(n)$.
- Pokiaľ $B[1]$ leží v kladnej polrovine určenej priamkou $A[1]A[m]$, potom uvažujeme hornú časť konvexného obalu množiny A a dolnú časť konvexného obalu množiny B , inak analogicky naopak. Ak uvedené dvojice mnohoúhelníkov majú spoločný bod, potom množiny A a B nie sú separovateľné, inak sú separovateľné. Táto časť algoritmu vyžaduje lineárny čas vzhľadom k počtu vrcholov v oboch mnohoúhelníkoch, t.j. $O(m+n)$. Jedným prechodom takto vzniknutej postupnosti bodov zistíme možné priesečníky úsečiek (vďaka usporiadaniu bodov podľa x -ových súradníc).
- Predpokladajme naďalej, že $B[1]$ leží v kladnej polrovine určenej priamkou $A[1]A[m]$. Ak sú množiny separovateľné, potom podľa vyššie uvedených predpokladov očakávame, že množina B bude celá v kladnej polrovine a množina A v zápornej polrovine hľadanej priamky p . Zvoľme priamku $p : ax + by + c = 0$ kolmú na priamku určenú bodmi $A[1]$ a $B[1]$. Zrejme platí, že body $A[1]$ a $B[1]$ sú separované správne. Ďalšie kroky budú spočívať vo vhodnej adaptácii tejto priamky. Prechodom všetkými bodmi oboch množín (stačia len body odpovedajúcich častí konvexného obalu) dostaneme správne nastavenú priamku. Ak je ďalší skúmaný bod zaradený správne, nič sa nedeje. Ak je bod v nesprávnej polrovine, je potrebné upraviť polohu priamky. To možno previesť pomocou nasledujúceho pravidla: Nech bod C je zaradený nesprávne, potom

$$\begin{aligned} da &:= 2 * eta * k * C.x, \\ db &:= 2 * eta * k * C.y, \\ dc &:= 2 * eta * k, \end{aligned}$$

kde $k = 1$ pre body množiny B , $k = -1$ pre body množiny A , eta je konštanta, $0 < eta < 1$. Nová rovnica priamky má potom tvar

$$(a + da)x + (b + db)y + (c + dc) = 0.$$

Tento krok vyžaduje lineárny čas vzhľadom na počet bodov oboch množín, teda $O(m+n)$.

P – I – 3

Zo zadania úlohy je zrejmé, že bude vhodné spočítať si najlepšie „nepriame“ výmenné kurzy dopredu (fáza predspracovania – zložitosť závisí od N , ale nie od P). Program potom môže ľahko odpovedať na otázky (zložitosť závisí od P , ale nie od N). Zavedieme preto (trojuholníkovú) tabuľku najlepších nepriamych výmenných kurzov $T[I, J]$ ($1 \leq I < J \leq N$), ktorá bude výsledkom predspracovania. Najprv popíšeme, čo vlastne znamená, že „najlepší nepriamy výmenný kurz medzi menami I a J je $T[I, J]$ “

a pokúsime sa nájsť vzťahy medzi prvkami $T[I, J]$ našej tabuľky, ktoré nám umožnia vytvoriť algoritmus.

Výmenou budeme nazývať postupnosť $V = (I_0, I_1, \dots, I_h)$ mien s vlastnosťou $I_0 < I_1 < \dots < I_h$, $h \geq 1$. Táto výmena realizuje (nepriamy) prevod meny I_0 na menu I_h prostredníctvom mien I_1, \dots, I_{h-1} (pre $h = 1$ máme samozrejme priamy prevod). Za jednu jednotku meny I_0 dostaneme po výmene V zrejme

$$K[I_0, I_1] \cdot K[I_1, I_2] \cdot \dots \cdot K[I_{h-1}, I_h]$$

jednotiek meny I_h – nazvime toto číslo *hodnotou* $H(V)$ výmeny V .

Najlepší výmenný kurz $T[I, J]$ medzi menami I a J teraz môžeme popísať ako maximálnu hodnotu spomedzi hodnôt všetkých výmen realizujúcich prevod meny I na menu J :

$$T[I, J] = \max \{H(V); V = (I_0, I_1, \dots, I_h), I = I_0, J = I_h\}.$$

Základným pozorovaním pre riešenie úlohy je nasledujúce tvrdenie (princíp optimality podvýmen): Ak nejaká výmena realizuje najlepší výmenný kurz (nepriamy prevod) medzi menami I a J a v tejto výmene bola použitá mena M ($I < M < J$), potom výmena I na M , resp. M na J , musela byť tiež optimálna – použitím lepšej výmeny by sme vylepšili aj výmenu I na J . Formálne zapísané teda platí:

Nech výmena $V = (I, \dots, M, \dots, J)$ má optimálnu hodnotu $H(V) = T[I, J]$. Označme si $V_1 = (I, \dots, M)$, $V_2 = (M, \dots, J)$. Potom $T[I, M] = H(V_1)$ a $T[M, J] = H(V_2)$, čiže tieto „podvýmeny“ sú optimálne a samozrejme $H(V) = H(V_1) \cdot H(V_2)$.

Myšlienka algoritmu by teraz už mala byť jasná – budeme určovať najlepšie výmenné kurzy medzi menami I a J postupne pre $J - I = 1, 2, 3, \dots, N - 1$. Hodnoty pre $J - I = 1$ sú samozrejme priame kurzy $K[I, J]$. Keď už poznáme hodnoty $T[I, J]$ pre $J - I < Q$, určíme $T[I, J]$ pre $J - I = Q$ ako minimálnu hodnotu spomedzi hodnôt

$$K[I, J], \quad T[I, I + 1] \cdot T[I + 1, J], \\ T[I, I + 2] \cdot T[I + 2, J], \quad \dots, \quad T[I, J - 1] \cdot T[J - 1, J].$$

Správnosť tohoto postupu zaručuje princíp matematickej indukcie a vyššie uvedený princíp optimality podvýmen.

Program môžeme vylepšiť ešte tým, že miesto dvoch tabuliek K a T budeme používať len jednu – dvojrozmerné pole *Kurzy*. V tomto poli budú na začiatku zapísané hodnoty z tabuľky K a postupne ich budeme nahradzovať vypočítanými hodnotami z tabuľky T . Prvky T počítame po uhlopriečkach v tomto poradí:

$$T[1, 2], \quad T[2, 3], \quad \dots, \quad T[N - 1, N], \\ T[1, 3], \quad T[2, 4], \quad \dots, \quad T[N - 2, N], \\ T[1, 4], \quad T[2, 5], \quad \dots, \quad T[N - 3, N], \\ \dots, \quad T[1, N].$$

Pri výpočte hodnoty $T[I, J]$, ktorá sa potom uloží do $Kurzy[I, J]$, použijeme hodnoty $T[I, M]$ a $T[M, J]$, uložené v poli $Kurzy[I, M]$, resp. $Kurzy[M, J]$ a hodnotu $K[I, J]$ uloženú v $Kurzy[I, J]$.

P – I – 4

Táto úloha nie je algoritmicky príliš náročná. Stačí si uvedomiť, že potrebujeme len vypočítať hodnoty výstupov hradiel H_1 až H_k (tie už jednoznačne určujú hodnoty výstupov obvodu) a na to stačí jediný „prechod“ usporiadanou množinou hradiel H_1, H_2, \dots, H_k , lebo hodnota výstupu hradla závisí len od hodnôt vstupov obvodu príp. od hodnôt výstupov hradiel s menším poradovým číslom. Presnejšie:

- hodnotu výstupu hradla H_i , ($1 \leq i \leq k$) určuje typ hradla (NAND, NOR, NOT) a dvojice hodnôt jeho vstupov. Každý vstup hradla H_i je pritom buď vstupom obvodu (I_1, I_2, \dots, I_n), alebo výstupom hradla H_j , $j < i$.
- hodnota výstupu obvodu O_i , ($1 \leq i \leq m$) sa rovná buď hodnote niektorého vstupu obvodu (I_1, I_2, \dots, I_n), alebo hodnote výstupu niektorého hradla (H_1, H_2, \dots, H_k).

Zložitosť algoritmu je zrejme lineárna vzhľadom na $n + k + m$.

Zaujímavejšia je otázka vnútornej reprezentácie obvodu a formátu textového popisu obvodu. Pekné (ale pre dané účely zbytočne komplikované) riešenie by sme mohli zapísať objektovými jazykovými prostriedkami. Uvedieme tu jednoduchšie riešenie v jazyku Turbo Pascal, ktoré objektové rysy jazyka nepoužíva.

Vnútorná reprezentácia obvodu:

Jednotlivé hradlá budeme reprezentovať záznamom, celý obvod bude uložený ako pole týchto záznamov. Záznam typu *Hradlo* obsahuje typ hradla (okrem NAND, NOR a NOT tiež „falošné“ hradlá typu IN a OUT – vstupy a výstupy obvodu), indexy *Vstup1*, *Vstup2* do poľa hradiel určujúce vstupy hradla a položku *Vystup* s možnými hodnotami 0, 1 pre hodnotu výstupu hradla. Pre hradlá typu IN je hodnota indexov *Vstup1*, *Vstup2* nezaujímavá a nastavuje sa na 0. Pre hradlá typu OUT hodnota indexu *Vstup1* určuje „napojenie“ daného výstupu obvodu, položka *Vstup2* sa nepoužíva.

Chovanie hradiel popisuje pole *CH*, ktoré pre každý typ hradla (NAND, NOR a NOT) obsahuje tabuľku príslušnej boolovskej funkcie. Logické hodnoty môžeme reprezentovať typom 0..1 (a nie boolean), ako je to obvyklé v logických obvodoch. Aby sme nemuseli rozlišovať hradlá s jedným a s dvoma vstupmi, všetky považujeme za dvojevstupové, pričom druhý vstup hradla typu NOT je stotožnený s jeho prvým vstupom – výpočet výstupnej hodnoty to neovplyvní.

Formát popisu obvodu:

Popis obvodu začína trojicou čísel N, K, M na samostatnom riadku udávajúcich po rade počet vstupov, hradiel a výstupov obvodu. Nasledujú popisy K hradiel a potom M výstupov (vstupy sa ďalej nepopisujú). Popis hradla a výstupu je tvaru:

NAND	<i>vstup1</i>	<i>vstup2</i>
NOR	<i>vstup1</i>	<i>vstup2</i>
NOT	<i>vstup1</i>	
OUT	<i>vstup1</i> ,	

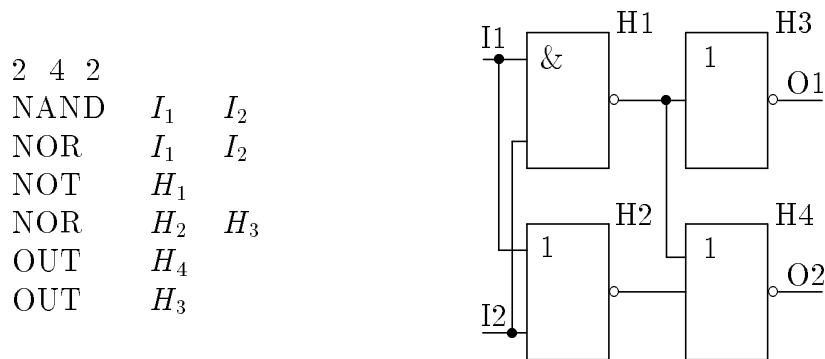
kde $vstup1$ a $vstup2$ má jeden z dvoch možných tvarov:

- I_i ($1 \leq i \leq N$) – daný vstup je i -tým vstupom obvodu,
- H_i ($1 \leq i \leq K$) – daný vstup je výstupom i -teho hradla.

V popise sa môžu vyskytovať medzery, tabulátory a znaky konca riadku na ľubovoľnom mieste (okrem „kľúčových slov“ NAND, NOR, NOT, OUT). Popis musí vyhovovať podmienkam úlohy: v popise i -teho hradla sa nesmie vyskytnúť odvolanie na hradlo H_j , pre ktoré $j \geq i$.

Riešenie časti b):

Je jasné, že výstup c sa dá spočítať ako NOT (x NAND y). Všimnite si, že c OR (x NOR y) je presne NOT s , teda s vypočítame ako c NOR (x NOR y). Uvádzame ešte schému a zápis v navrhnutom formáte popisu obvodu.



P – II – 1

Dôležitou charakteristikou situácie hry je počet *inverzií*: dvojíc kamienkov modrý–červený, kde modrý kamienok (budeme ho značiť m) je naľavo od červeného kamienka (c). Je jasné, že počet inverzií koncovej situácie je 0 (lebo všetky červené kamienky sú naľavo od modrých). Maximálny počet inverzií (n^2) nastáva pri situácii

$$\underbrace{mm \dots m}_n \underbrace{cc \dots c}_n .$$

Ak urobíme krok v hre (obrátenie poradia troch alebo štyroch susedných kamienkov), zmeníme len vzájomné poradie obrátených kamienkov a inverzie medzi nimi a ostatnými kamienkami ostanú zachované. Preto celkový počet inverzií sa zmenší (príp. zväčší) nanajvýš o 4. Ak máme situáciu s p inverziami, potrebujeme aspoň $\frac{p}{4}$ krokov na to, aby sme ju previedli na koncovú. V najhoršom prípade to znamená kvadratický počet krokov $\left(\frac{n^2}{4}\right)$.

Nech p je pozícia prvého červeného kamienka (počítame zľava), naľavo od ktorého je modrý kamienok. Takúto dvojicu kamienkov (mc) nazveme *prechodovou dvojicou*,

pozíciu p nazveme *prechodom*. Náš algoritmus spočíva v tom, že nájdeme pozíciu p , potom pomocou jedného alebo najviac dvoch ťahov znížime počet inverzií aspoň o 1 (vychádza teda zníženie inverzie aspoň o $\frac{1}{2}$ na ťah, a teda počet ťahov bude kvadratický) a opäť nájdeme novú pozíciu prechodu. V prípade, že už neexistuje prechodová dvojica, dosiahli sme koncovú situáciu.

Podľa toho, aké kamienky sú v susedstve prechodovej dvojice, je potrebné rozlíšiť niekoľko situácií. Pri každej z nich určíme, aký krok je treba vykonať a od ktorého miesta treba začať hľadať novú prechodovú pozíciu. Nech funkcia $UrobKrok(p, k)$ realizuje krok (p, k) . Prechod znázorníme znakom $|$ umiestneným medzi prechodovú dvojicu $(m|c)$. Okraj hracieho plánu znázorníme znakom \langle , resp. \rangle a miesto, kde je potrebné začať hľadať novú prechodovú dvojicu označíme znakom \bullet .

$cm cc$	\rightarrow	$ccc\bullet$	$UrobKrok(p - 1, 3)$
$\langle m cc$	\rightarrow	$\langle cc\bullet$	$UrobKrok(p - 1, 3)$
$cm cmc$	\rightarrow	$cc\bullet cm$	$UrobKrok(p - 1, 4)$
$\langle m cmc$	\rightarrow	$\langle c\bullet cm$	$UrobKrok(p - 1, 4)$
$cm cmm$	\rightarrow	$cmmmc \rightarrow cmm\bullet$	$UrobKrok(p, 3), UrobKrok(p - 1, 4)$
$\langle m cmm$	\rightarrow	$\langle mmmc \rightarrow \langle cmm\bullet$	$UrobKrok(p, 3), UrobKrok(p - 1, 4)$
$mmm cc$	\rightarrow	$\bullet ccmm$	$UrobKrok(p - 2, 4)$
$cmm cc$	\rightarrow	$cccm\bullet$	$UrobKrok(p - 2, 4)$
$\langle mm cc$	\rightarrow	$\langle ccm\bullet$	$UrobKrok(p - 2, 4)$
$mmm cm$	\rightarrow	$\bullet cmmm$	$UrobKrok(p - 2, 3)$
$cmm cm$	\rightarrow	$ccm\bullet m$	$UrobKrok(p - 2, 3)$
$\langle mm cm$	\rightarrow	$\langle cm\bullet m$	$UrobKrok(p - 2, 3)$
$mmm c\rangle$	\rightarrow	$\bullet cmm\rangle$	$UrobKrok(p - 2, 3)$
$cmm c\rangle$	\rightarrow	$ccm\bullet\rangle$	$UrobKrok(p - 2, 3)$
$\langle mm c\rangle$	\rightarrow	$\langle cm\bullet\rangle$	$UrobKrok(p - 2, 3)$

Starostlivé sledovanie pozície nového prechodu zabezpečí, aby nielen počet krokov hry, ale aj časová zložitosť algoritmu bola kvadratická. Keďže v najhoršom prípade môže byť aj počet potrebných ťahov kvadratický, tento algoritmus je asymptoticky optimálny.

Nová prechodová dvojica však niekedy môže byť veľmi vzdialená a pri jej hľadaní je potrebné pozrieť až rádovo n kamienkov (nazvime túto situáciu *dlhé hľadanie*). Každé dlhé hľadanie sa začína v situácii $c\bullet$, $cm\bullet$, alebo $cmm\bullet$ na pozícii \bullet a pokračuje cez niekoľko modrých kamienkov, kým nenájde červený. Máme teda prípad tvaru

$$c \underbrace{mmmm \dots m}_j c,$$

kde $j \geq 3$). Prechod v takomto prípade nájdeme pomocou $O(j)$ krokov. Pri ďalších zhruba $\frac{j}{2}$ ťahoch sa nájdený červený kamienok začne sťahovať doľava cez j modrých

kamienkov a počet inverzií sa postupne zníži zhruba o j . Počas týchto ťahov nájdeme pozíciu ďalšej prechodovej dvojice v konštantnom čase. To znamená, že dlhému hľadaniu ($O(j)$ krokov programu) môžeme priradiť následné zníženie počtu inverzií o $O(j)$ v čase $O(j)$, čo ukazuje, že celkový počet krokov programu strávených v dlhých hľadániach je $O(n^2)$.

P – II – 2

Z daného bodu môže robot pokračovať tromi smermi, označme ich a, b, c . Prechodom po prvej ceste a zaznamenaním typu smeru pre každú úsečku dostaneme reťazec A_r charakterizujúci cestu. Analogicky získame reťazec B_r . Oba prechody je možné urobiť v lineárnom čase $O(n)$, resp. $O(m)$.

Problém sa tak redukuje na určovanie dĺžky najdlhšej spoločnej podpostupnosti dvoch reťazcov. Algoritmus má zložitosť $O(mn)$ a je založený na nasledujúcom princípe: Budeme postupne počítat hodnoty matice SP, kde $SP[i, j]$ bude obsahovať dĺžku najdlhšej spoločnej podpostupnosti pre podreťazce $A_r[1] \dots A_r[i]$ a $B_r[1] \dots B_r[j]$.

Ak chceme vypočítat dĺžku najdlhšej spoločnej podpostupnosti dvojice podreťazcov $A_r[1] \dots A_r[i]$, $B_r[1] \dots B_r[j]$, potrebujeme poznať hodnoty $SP[i-1, j-1]$, $SP[i, j-1]$ a $SP[i-1, j]$. Pri výpočte budeme rozlišovať, či sa prvky $A_r[i]$ a $B_r[j]$ rovnajú alebo nie. Ak platí $A_r[i] = B_r[j]$, potom $SP[i, j] = SP[i-1, j-1] + 1$. Ak platí $A_r[i] \neq B_r[j]$, potom $SP[i, j] = \max\{SP[i, j-1], SP[i-1, j]\}$.

Celý výpočet je potrebné urobiť pre $i = 1, \dots, m$ a $j = 1, \dots, n$. Pomocné hodnoty, ak je niektorý z podreťazcov prázdny, je potrebné stanoviť na nulu. Po určení dĺžky najdlhšej spoločnej podpostupnosti A_r a B_r získame hľadané počty jej odpočítaním od dĺžok ciest.

P – II – 3

Riešenie tejto úlohy je podobné riešeniu úlohy P-I-3. Jediný rozdiel je v tom, že pre každú dvojicu štátov (i, j) , kde $i < j$, určíme 11 čísel: optimálne výmenné kurzy pri maximálnom zdržaní $0, 1, 2, \dots, 10$ dní. Myšlienka algoritmu je jednoduchá: ak cestujeme z i do j a máme povolené zdržanie k dní, optimálny kurz dosiahneme ako maximum nasledujúcich čísel:

- optimálny kurz z i do j bez medzistaníc (a teda bez zdržania),
- maximum z kurzov získateľných so zastávkou v medzistanici l pre všetky medzistanice l ($i < l < j$) a pre všetky možné „rozklady“ zdržania k na tri nezáporné časti: zdržanie medzi i a l , zdržanie v l (hodnota $T[l]$) a zdržanie medzi l a j .

Počet rozkladov čísla k je zhora ohraničený konštantou (lebo maximálne možné zdržanie je 10), a preto je časová zložitosť celého algoritmu $O(n^3)$.

P – II – 4

Jedna možnosť, ako vytvoriť prehľadný obvod na riešenie úlohy, je rozložiť obvod na dve časti:

- **dekodér**, ktorý zo 4 vstupov x_0, x_1, x_2, x_3 vytvorí 10 medzivýstupov z_0, z_1, \dots, z_9 , kde je z_i má hodnotu 0 (*false*) vtedy a len vtedy, ak $x_3x_2x_1x_0$ tvorí binárny zápis čísla i (zvolili sme *false* a nie *true* z dôvodu lepšej väzby na kodér).
- **kodér**, ktorý bude realizovať funkcie typu „ľavý horný segment má svietiť, ak $x_3x_2x_1x_0$ tvorí binárny zápis čísla 0, 4, 5, 6, 8 alebo 9“. Inak povedané to znamená, že $y_1 = \overline{z_0} \vee \overline{z_4} \vee \overline{z_5} \vee \overline{z_6} \vee \overline{z_8} \vee \overline{z_9} = \neg(z_0 \wedge z_4 \wedge z_5 \wedge z_6 \wedge z_8 \wedge z_9) = \text{NAND}(z_0, z_4, z_5, z_6, z_8, z_9)$, teda kodér zo vstupov z_0, z_1, \dots, z_9 vytvorí výstupy $y_0, y_1, y_2, y_3, y_4, y_5, y_6$ len s použitím hradiel typu NAND (preto boli medzivýsledky „negované“).

Dekodér: Začneme tým, že vytvoríme negácie $\overline{x_0}, \overline{x_1}, \overline{x_2}, \overline{x_3}$ vstupov x_0, x_1, x_2, x_3 , teda 4 hradlá typu NOT :

$$\begin{aligned} H1 &= \text{NOT}(I1) = \overline{x_0} \\ H2 &= \text{NOT}(I2) = \overline{x_1} \\ H3 &= \text{NOT}(I3) = \overline{x_2} \\ H4 &= \text{NOT}(I4) = \overline{x_3} \end{aligned}$$

Nasledujúcich 10 hradiel typu NAND so 4 vstupmi vytvorí požadované výstupy z_0, z_1, \dots, z_9 :

$$\begin{aligned} H5 &= \text{NAND}(H1, H2, H3, H4) = \neg(\overline{x_0} \wedge \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3}) = z_0 \\ H6 &= \text{NAND}(I1, H2, H3, H4) = \neg(x_0 \wedge \overline{x_1} \wedge \overline{x_2} \wedge \overline{x_3}) = z_1 \\ H7 &= \text{NAND}(H1, I2, H3, H4) = \neg(\overline{x_0} \wedge x_1 \wedge \overline{x_2} \wedge \overline{x_3}) = z_2 \\ H8 &= \text{NAND}(I1, I2, H3, H4) = \neg(x_0 \wedge x_1 \wedge \overline{x_2} \wedge \overline{x_3}) = z_3 \\ H9 &= \text{NAND}(H1, H2, I3, H4) = \neg(\overline{x_0} \wedge \overline{x_1} \wedge x_2 \wedge \overline{x_3}) = z_4 \\ H10 &= \text{NAND}(I1, H2, I3, H4) = \neg(x_0 \wedge \overline{x_1} \wedge x_2 \wedge \overline{x_3}) = z_5 \\ H11 &= \text{NAND}(H1, I2, I3, H4) = \neg(\overline{x_0} \wedge x_1 \wedge x_2 \wedge \overline{x_3}) = z_6 \\ H12 &= \text{NAND}(I1, I2, I3, H4) = \neg(x_0 \wedge x_1 \wedge x_2 \wedge \overline{x_3}) = z_7 \\ H13 &= \text{NAND}(H1, H2, H3, I4) = \neg(\overline{x_0} \wedge \overline{x_1} \wedge \overline{x_2} \wedge x_3) = z_8 \\ H14 &= \text{NAND}(I1, H2, H3, I4) = \neg(x_0 \wedge \overline{x_1} \wedge \overline{x_2} \wedge x_3) = z_9 \end{aligned}$$

Kodér: Sedem hradiel kodéra realizuje funkcie jednotlivých segmentov displeja tak, ako sme popísali vyššie. Všetky hradlá sú typu NAND, majú 4 až 9 vstupov spomedzi $H5, \dots, H14$:

$$\begin{aligned} H15 &= \text{NAND}(H5, H7, H8, H10, H11, H12, H13, H14) &= y_0 \\ H16 &= \text{NAND}(H5, H9, H10, H11, H13, H14) &= y_1 \\ H17 &= \text{NAND}(H5, H6, H7, H8, H9, H12, H13, H14) &= y_2 \\ H18 &= \text{NAND}(H7, H8, H9, H10, H11, H13, H14) &= y_3 \\ H19 &= \text{NAND}(H5, H7, H11, H13) &= y_4 \\ H20 &= \text{NAND}(H5, H6, H8, H9, H10, H11, H12, H13, H14) &= y_5 \\ H21 &= \text{NAND}(H5, H7, H8, H10, H11, H13, H14) &= y_6 \end{aligned}$$

P – III – 1

Pre každú množinu A obsahujúcu n modrých a n červených bodov spĺňajúcu zadanie, existuje riešenie. Dôkazom bude algoritmus, ktorý pre každú takúto množinu zostrojí riešenie v čase $O(n^2)$.

Usporiadajme najprv body množiny A do postupnosti $(x_1, y_1), (x_2, y_2), \dots, (x_{2n}, y_{2n})$ tak, že pre $i < j$ platí $x_i \leq x_j$, a ak $x_i = x_j$, potom $y_i < y_j$. Toto je možné spraviť v čase $O(n \log n)$.

Potom rozdelíme postupnosť A na k disjunktných súvislých neprázdnych úsekov tak, aby v každom úseku bol rovnaký počet červených a modrých bodov, ale v žiadnom kratšom úseku začínajúcom na tom istom mieste nie. Úseky si budeme ukladať v zozname Z . Túto časť výpočtu je možné vykonať v čase $O(n)$ jedným prechodom postupnosti A . V každom úseku je rovnaký počet červených a modrých bodov, preto úlohu vyriešime zvlášť pre každý úsek. Konvexné obaly jednotlivých úsekov sa neprekrývajú a úsečky spájajúce dva body z jedného úseku ležia vnútri konvexného obalu tohto úseku. Preto sa úsečky spájajúce body v rôznych úsekoch nebudú pretínať a riešenie pre celú množinu A dostaneme jednoducho spojením riešení pre jednotlivé úseky.

Spracovávame teda postupne jednotlivé úseky zo zoznamu Z (vždy vyberieme jeden úsek U zo zoznamu). Lahko možno ukázať, že prvý a posledný bod úseku majú rôznu farbu. Prvý a posledný bod úseku sú určite vrcholy mnohouholníka tvoriaceho konvexný obal bodov úseku, a teda v tomto mnohouholníku existujú určite aj dva za sebou idúce vrcholy, ktoré majú rôznu farbu. Nech U' je úsek, ktorý vznikne z U odstránením týchto dvoch vrcholov. Konvexný obal U' nemá žiaden spoločný bod s úsečkou spájajúcou dva odstránené body. Preto ak nájdeme riešenie U' a pridáme k nemu túto úsečku, dostávame riešenie pre U . Postupujeme teda nasledovne: zostrojíme konvexný obal U , nájdeme na ňom dva susedné body rôznej farby, tieto spojíme úsečkou a vyhodíme z U . Zo zvyšných bodov opäť vytvoríme úseky rovnakým spôsobom ako z pôvodnej postupnosti bodov a zaradíme ich do Z . Takto sme celkový počet bodov znížili o 2, a teda po n takýchto krokoch už bude zoznam Z prázdny a nájsť úsečky tvoria riešenie úlohy.

Konvexný obal množiny m bodov usporiadanej podľa x -ovej súradnice je možné spraviť v čase $O(m)$ nasledujúcim algoritmom: najľavejší a najpravejší bod množiny (označme ich L a P) sú iste vrcholy konvexného obalu. Ostatné body množiny rozdelíme na dve skupiny podľa toho, či ležia pod alebo nad priamkou LP . Teraz môžeme zvlášť nájsť hornú časť konvexného obalu a zvlášť jeho dolnú časť.

Dolnú časť konvexného obalu budeme hľadať tak, že budeme postupne zľava doprava pridávať jednotlivé body ležiace pod priamkou LP do zoznamu vrcholov konvexného obalu. Vždy keď pridáme ďalší bod, skontrolujeme, či posledné tri body v zozname netvoria nekonvexný uhol. Ak tvoria, predposledný bod zoznamu je potrebné zo zoznamu vyhodit' a opäť skontrolovať posledné tri body v zozname. To robíme dovtedy, kým uhol na konci zoznamu nebude konvexný, alebo kým v zozname nezostanú iba dva body. Potom môžeme pridať ďalší bod. Takto postupujeme, až kým nespracujeme všetky body v danej polrovine, vrátane bodu P . Horná časť konvexného obalu sa vytvorí analogicky.

Každý vrchol pridáme do zásobníka práve raz a najviac raz ho odtiaľ odstránime. Obidve operácie vyžadujú čas $O(1)$, a preto konvexný obal m bodov je možné zostrojiť v čase $O(m)$. Keď máme zostrojený konvexný obal, je možné v čase $O(n)$ nájsť dvojicu susedných vrcholov rôznej farby, odstrániť ich z poľa a zo zvyšných bodov úseku vytvoriť nové úseky.

Spracovanie jedného úseku teda vieme vykonať v čase $O(n)$. Spracovaním každého úseku spojíme úsečkou jednu dvojicu bodov. Keďže potrebujeme pospájať n dvojíc bodov, výsledný čas je $O(n^2)$.

P – III – 2

Predpokladajme, že máme danú ľubovoľnú schodovitú postupnosť $S = A[1], A[2], \dots, A[n] = K$. Podľa zadania sa vektor $A[i]A[i+1]$, $1 \leq i < n$ musí zhodovať s jedným z vektorov $\vec{a} = (2, -1)$, $\vec{b} = (1, -1)$, alebo $\vec{c} = (1, -2)$. Takže postupnosť $A[1], A[2], \dots, A[n]$ je jednoznačne určená bodom S a postupnosťou vektorov v_1, v_2, \dots, v_n , kde $v_i \in \{\vec{a}, \vec{b}, \vec{c}\}$. Keďže sčítanie vektorov je komutatívne, ľubovoľná permutácia prvkov postupnosti v_1, v_2, \dots, v_n je reprezentáciou vyhovujúcej schodovitej postupnosti začínajúcej bodom S a končiacej bodom K . Stačí si teda pamätať počet vektorov \vec{a} , \vec{b} a \vec{c} v postupnosti $(v_i)_{i=1}^n$.

Ak by sa v postupnosti $(v_i)_{i=1}^n$ nachádzali aspoň tri vektory \vec{b} , vedeli by sme ich nahradiť kombináciou vektorov \vec{a} , \vec{c} , a tak znížiť počet bodov nájdenej schodovitej postupnosti. Minimálna schodovitá postupnosť teda obsahuje najviac dva vektory \vec{b} .

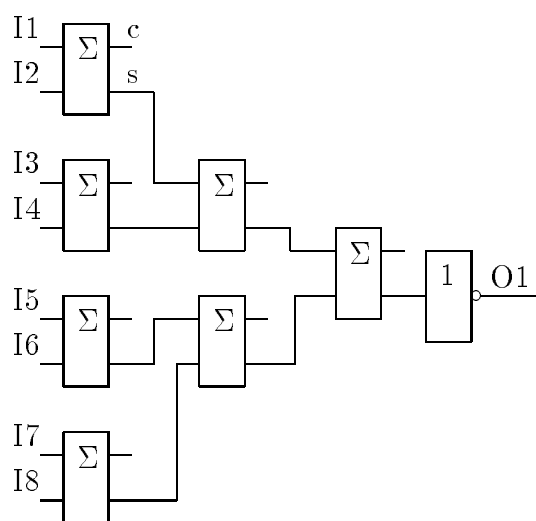
Zrejme ak $K.x - S.x < 2(S.y - K.y)$ alebo $S.y - K.y < 2(K.x - S.x)$, riešenie neexistuje. V opačnom prípade riešenie vždy existuje.

Predstavme si obdĺžnik $SVKW$, pričom hrany SV , KW sú vodorovné a SW , KV zvislé. Môžu nastať tri prípady:

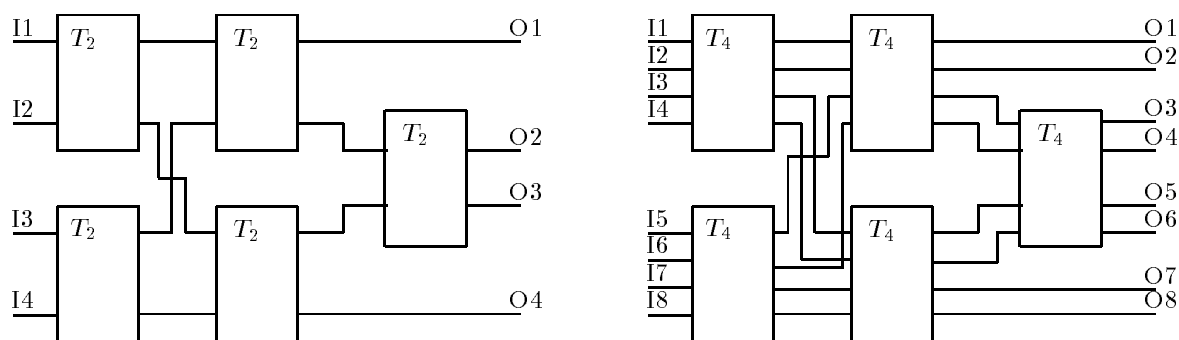
- $|SV| > |SW|$ – v schodovitej postupnosti sa musí nachádzať aspoň jeden vektor \vec{a} . Posunieme bod S v smere \vec{a} do bodu S' a hľadáme minimálnu schodovitú postupnosť pre body S' a K . Táto bude spolu s bodom S tvoriť minimálnu schodovitú postupnosť pre body S a K .
- $|SV| < |SW|$ – analogicky predošlému prípadu, ale bod S posunieme o vektor \vec{c} .
- $|SV| = |SW|$ – obdĺžnik $SVKW$ je štvorcem. Ak $|SV| \geq 3$, budeme S namiesto posúvania o vektor \vec{b} posúvať o vektory \vec{a} a \vec{c} (už sme ukázali, že je to najvýhodnejšie). Toto pravidlo nemôžeme použiť, ak $|SV| \leq 2$. Vtedy musíme postupnosť zakončiť „schodmi“ – vektormi \vec{b} .

P – III – 3

Funkcia *parita*: Na riešenie tejto úlohy využijeme obvod, ktorý sme zostavili už v I. kole (pozri obrázok na strane 19). Obvod mal dva vstupy $I1, I2$ a dva výstupy c, s , pričom platilo, že $2c + s = I1 + I2$. Tento obvod využijeme na sčítanie všetkých vstupov modulo 2 takto: najprv sčítame vstupy po dvojiciach, čím dostávame štyri súčty dvojíc modulo 2 a štyri zvyškové bity. Zvyškové bity nás ďalej nebudú zaujímať, dvojice súčtov opäť sčítame tým istým spôsobom. Dostaneme dva súčty modulo 2 (prvých štyroch a druhých štyroch bitov), ktoré opäť sčítame. Výsledok musíme ešte znegovať, keďže náš obvod dáva v prípade párneho počtu jedničiek medzi vstupmi výsledok 0.



Funkcie majorita, aspoň2: Zostavme najprv pomocný obvod (budeme ho označovať T_k , kde k je počet jeho vstupov), ktorý dostane k vstupov núl alebo jednotiek a na výstupoch ich vráti utriedené (t.j. najprv všetky nuly a potom všetky jednotky). Obvod T_2 bol uvedený ako príklad v študijnom texte (pozri str. 3). Pokúsme sa teraz zostrojiť obvod T_4 . Pomocou troch obvodov T_2 vieme zo vstupných štyroch bitov (I_1, \dots, I_4) získať najmenší – vezmeme menší z I_1, I_2 a menší z I_3, I_4 a pomocou ďalšieho obvodu T_2 ich porovnáme. Podobne vieme nájsť najväčší bit. Zostávajú dva bity, ktoré pomocou jedného obvodu T_2 možno utriediť. (Výsledný obvod je na obrázku vľavo.) Zostrojiť obvod T_8 pomocou obvodov T_4 teraz už nebude problém. Stačí zopakovať predchádzajúci postup, teraz však už uvažujeme dvojice bitov (obrázok vpravo).



Pôvodne požadované obvody zostrojíme teraz pomocou obvodu T_8 : stačí vybrať príslušný výstup tohto obvodu – v prípade majority to bude O_4 a v prípade aspoň2 O_7 .

P – III – 4

Stačí sa zaoberať hrou so 4 kameňkami. Ostatné prípady je totiž možné previesť na tento nasledovne: rozšírime hrací plán zľava o 4 – početKamienkov políček, na ktoré

umiestnime kamienky, ktoré sa už samozrejme nepohnú. Jednotlivé situácie v hre budeme charakterizovať hodnotami m_1 a m_2 , pričom pre pozície kamienkov x, y, u, v ($x < y < u < v$) definujeme $m_1 = y - x - 1$, $m_2 = v - u - 1$.

Platia nasledujúce tvrdenia:

1. Ak $m_1 \neq m_2$, hráč, ktorý je na ťahu, môže urobiť krok, ktorým postaví druhého hráča do situácie $m_1 = m_2$.

Dôkaz: Hráč na ťahu môže urobiť krok dĺžky $|m_1 - m_2|$ kamienkom na pozícii y (ak $m_1 > m_2$) alebo kamienkom na pozícii v (ak $m_1 < m_2$).

2. Ak $m_1 = m_2$, hráč, ktorý je na ťahu nemôže zabrániť výhre druhého hráča.

Dôkaz: Použijeme matematickú indukciu vzhľadom na súčet $S = x + y + u + v$. Ak sme v koncovej situácii (t.j. $S = 6$ pre prípad so štyrmi kamienkami), hráč, ktorý je na ťahu, nemôže ťahať, a preto prehral. V nekoncovej situácii, v ktorej platí $m_1 = m_2$, hráč, ktorý je na ťahu svojim ťahom poruší platnosť tejto podmienky. Druhý hráč môže odpovedať ťahom podľa bodu 1, a tým zasa postaví prvého hráča do situácie $m_1 = m_2$, pričom súčet pozícií kamienkov S sa znížil a podľa indukčného predpokladu hráč na ťahu nemôže zabrániť svojej prehre.

Z dokázaných tvrdení vyplýva, že pri $m_1 \neq m_2$ má hráč, ktorý je na ťahu, vyhrávajúcu stratégiu, v opačnom prípade nemôže vyhrať, ak spoluhráč neurobí chybu.

P – III – 5

Situáciu môžeme reprezentovať ako graf G , ktorého vrcholy budú rozdelené do dvoch disjunktných množín. V množine A budú vrcholy zodpovedajúce poradovým číslam dverí a v množine B vrcholy zodpovedajúce jednotlivým dverám. Vrchol $u \in A$ a vrchol $v \in B$ budú spojené hranou v prípade, ak dané podmienky nezakazujú, aby dvere v boli otvorené ako u -te v poradí. *Párovanie* v grafe je taká podmnožina hrán, že žiadne dve z nich nemajú spoločný vrchol. Párovanie M pokrýva vrchol grafu v , ak nejaká hrana z M vychádza z vrcholu v . Úlohou je teda vlastne nájsť párovanie, ktoré bude pokrývať všetky vrcholy z A . Keďže $|A| = |B|$ a všetky hrany vedú iba z A do B , znamená to, že takéto párovanie pokrýva aj všetky vrcholy z B .

Stupeň vrcholu v z množiny A môže byť 0, 1, 2 (ak sa na v vzťahuje nejaká podmienka), alebo n (ak sa na v nevzťahuje žiadna podmienka). Ak má niektorý vrchol stupeň 0, žiadne riešenie úlohy neexistuje. Ak existuje vrchol $v \in A$, z ktorého vychádza jediná hrana (do vrcholu $u \in B$), potom každé riešenie úlohy musí obsahovať hranu uv . Ak teda z grafu odstránime vrcholy u a v a všetky hrany z nich vychádzajúce, novovzniknutý graf bude mať rovnaký počet riešení ako ten pôvodný (ak zadefinujeme, že graf s 0 vrcholmi má práve jedno riešenie). Takto môžeme postupne odstraňovať vrcholy stupňa 1 z množiny A , pričom však treba dávať pozor na to, že odstránením jedného takého vrcholu môžu vzniknúť ďalšie.

Stačí teda uvažovať prípad, keď $n > 1$ a každý vrchol množiny A má stupeň aspoň 2. Nech A' je podmnožina množiny A obsahujúca iba vrcholy so stupňom 2 a nech G' je podgraf grafu G s vrcholmi $A' \cup B$ a všetkými hranami grafu G , ktoré idú z A' do B . Ak existuje párovanie pokrývajúce všetky vrcholy množiny A' , potom existuje aj párovanie pokrývajúce všetky prvky množiny A , pretože na vrcholy z množiny $A \setminus A'$

sa nevzťahuje žiadna podmienka, a preto ich môžeme ľubovoľne spárovať so zvyšnými vrcholmi množiny B . Zároveň však platí, že ak $|A \setminus A'| > 1$, potom úloha iste nemá práve jedno riešenie. Ak $|A \setminus A'| \in \{0, 1\}$, potom má úloha práve toľko riešení, koľko existuje párovaní v G' pokrývajúcej množinu A' .

Zostáva teda už iba vyriešiť problém, koľko existuje v grafe G' párovaní pokrývajúcej množinu A' . Ak $A' = \emptyset$ existuje práve jedno párovanie požadovaných vlastností, v opačnom prípade nemôže existovať jediné také párovanie. Nech existuje nejaké párovanie pokrývajúce množinu A' . Vytvoríme cestu obsahujúcu striedavo hrany obsiahnuté v párovaní a hrany, ktoré v párovaní nie sú. Začneme v nejakom vrchole $v_0 \in B$ pokrytom párovaním a z tohto vrcholu budeme pokračovať po hrane obsiahnutej v párovaní do vrcholu $v_1 \in A'$. Odtiaľ môžeme pokračovať jedine po druhej hrane z neho idúcej do nejakého vrcholu $v_2 \in B$. Odtiaľ znovu po hrane z párenia atď. Skončíme vtedy, keď prideme do vrcholu $v \in B$ stupňa 1, alebo ak sa vrátíme do niektorého vrcholu, v ktorom sme už boli. V prvom prípade je možné vytvoriť nové párovanie tak, že hrany ležiace na ceste, ktoré v pôvodnom párovaní neboli, tam zaradíme a tie, ktoré v pôvodnom párovaní boli, odstránime. V druhom prípade časť cesty vytvára kružnicu, na tejto kružnici tiež môžeme vymeniť hrany rovnakým spôsobom a získať nové párovanie. Teda ak existuje jedno párovanie pokrývajúce množinu A' , potom určite existujú aspoň dve takéto párovania.

V prípade, že $A' \neq \emptyset$, stačí zistiť, či vôbec nejaké párovanie existuje. Ak v množine B existuje vrchol v , ktorý je spojený hranou iba s jediným vrcholom $u \in A'$, potom ak existuje párovanie pokrývajúce A' , existuje aj také párovanie pokrývajúce A' , ktoré obsahuje hranu uv (vezmeme ľubovoľné párovanie pokrývajúce A' , odstránime z neho hranu obsahujúcu u a pridáme hranu uv). Teda môžeme z grafu odstrániť vrcholy u a v a všetky hrany z nich idúce a ďalej postupne odstraňovať ďalšie vrcholy množiny B spojené s jediným vrcholom množiny A' (opäť nám odstránením jedného môžu vzniknúť ďalšie takéto vrcholy).

Nech A'' je množina vrcholov obsahujúca vrcholy z A' , ktoré neboli odstránené v procese opísanom vyššie a B' nech je podmnožina množiny B obsahujúca vrcholy, ktoré sú spojené hranou s nejakým vrcholom z A'' . Keďže sme odstránili vrcholy, ktoré boli spojené iba s jedným vrcholom z A' , každý vrchol z B' je spojený s aspoň dvoma vrcholmi z A'' . Súčasne však každý vrchol z A'' je spojený práve s dvoma vrcholmi z B' . Počet hrán idúcich z A'' do B' musí byť rovnaký, ako počet hrán idúcich z B' do A'' , a preto $|A''| \geq |B'|$. Ak $|A''| > |B'|$, potom iste neexistuje párovanie pokrývajúce množinu A'' . Ak $|A''| = |B'|$, potom každý vrchol množiny B' má stupeň práve 2. Nech G'' je podgraf grafu G' s vrcholmi $A'' \cup B'$ a všetkými hranami grafu G' , ktoré idú z A'' do B' . Všetky komponenty súvislosti grafu G'' sú teda kružnice, v ktorých sa striedajú vrcholy z A'' a vrcholy z B' . Párovanie v tomto grafe vytvoríme jednoducho tak, že na každej takejto kružnici striedavo jednu hranu zaradíme do párovania a druhú nie.

Na základe týchto úvah je možné napísať algoritmus, ktorý bude riešiť úlohu v čase $O(n)$. Pre každý vrchol si budeme pamätať zoznam hrán, ktoré z neho vychádzajú. Vrcholov, na ktoré sa nevzťahuje žiadna podmienka, môže byť až $O(n)$, pričom z každého vychádza n hrán. Už vytvorenie takejto štruktúry by vyžadovalo

čas $O(n^2)$, pri týchto vrcholoch si nebudeme pamätať zoznam hrán, iba ich počet. Každá hrana sa vyskytuje v dvoch zoznamoch (pre oba svoje koncové vrcholy). Ak tieto dva výskyty prepojíme, je možné v čase $O(1)$ vymazať danú hranu z obidvoch zoznamov. Potom sledujeme jednotlivé podmienky uvedené v predchádzajúcich úvahách, a keď je to potrebné, odstraňujeme vrcholy stupňa 1 (pričom si udržiavame zásobník nájdených a dosiaľ nespracovaných vrcholov stupňa 1). Odstrániť z grafu môžeme najviac $O(n)$ hrán, každú v čase $O(1)$, takže celkový algoritmus má lineárnu časovú zložitosť.

Štvrtá stredoeurópska olympiáda v informatike

V dňoch 17. – 24. júla 1997 sa v meste Nowy Sacz v Poľsku konala 4. stredoeurópska olympiáda v informatike. Na podujatí sa zúčastnilo 54 súťažiacich zo 14 krajín sveta, konkrétne: Bielorusko, Estónsko, Holandsko, Chorvátsko, Juhoslávia, Litva, Lotyšsko, Maďarsko, Nemecko, Poľsko, Rumunsko, Slovensko, Ukrajina a USA. Družstvo Českej republiky sa tohto roku na olympiáde, žiaľ, nezúčastnilo vzhľadom k povodňovej situácii. Podujatie sa konalo pod záštitou prezidenta republiky Alexandra Kwasniewského a za výraznej finančnej podpory poľského ministerstva školstva.

Slovensko na súťaži reprezentovali štyria súťažiaci, ktorí boli vybratí na základe výsledkov celoštátneho kola a výberového sústredenia: *Vladimír Koutný*, *Richard Kráľovič*, *Dávid Pál*, všetci z 2. ročníka Gymnázia J. Hronca v Bratislave a *Ján Svoreň* zo 4. ročníka Gymnázia D. Tatarku v Poprade. Vedúcou slovenského družstva bola *Bronislava Brejová* z MFF UK v Bratislave. Na základe pozvania z poľskej strany sa navyše olympiády zúčastnili *Martin Pál* a *Tomáš Vinař* (obaja z MFF UK v Bratislave) ako členovia poroty.

Samotná súťaž pozostávala z dvoch súťažných dní. Každý súťažný deň súťažiaci riešili tri príklady v časovom limite 5 hodín. Riešenia boli hodnotené už tradičným spôsobom – kontrolou výstupov programov pre zadané vstupné dáta, pričom bol stanovený časový limit pre beh programu. Každý deň bolo možné získať 100 bodov.

Spolu bolo udelených 28 medailí, z toho 5 zlatých, 9 strieborných a 14 bronzových. Naši súťažiaci dosiahli nasledujúce výsledky:

Por.	Meno	1. deň	2. deň	Súčet	Medaila
9.	Ján Svoreň	70	68	138	strieborná
24.	Vladimír Koutný	51	53	104	bronzová
43.	Richard Kráľovič	40	28	68	–
45.	Dávid Pál	37	22	59	–

Slovensko sa v celkovom neoficiálnom hodnotení krajín umiestnilo na siedmom mieste. O niečo horšie výsledky ako po minulé roky boli zapríčinené tým, že na túto súťaž boli vybraní súťažiaci nižších ročníkov, ktorí nemajú ešte žiadne skúsenosti s medzinárodnými súťažami. Účasť na tomto podujatí bola pre nich prípravou pre ich pravdepodobné účinkovanie na Medzinárodnej olympiáde z informatiky v budúcich rokoch.

Pre účastníkov bolo okrem samotnej súťaže pripravené množstvo ďalších programov, ako napríklad výlety do mesta Krakow a do historickej soľnej bane Wieliczka, prehliadka mesta Sacz, plavba na pltiach po Dunaji, alebo rozlúčková party.

Celé podujatie, ako odborná tak aj neodborná časť, bolo veľmi dobre zorganizované. Kvalitná príprava sa prejavila hlavne v až prekvapujúco hladkom priebehu vyhodnocovacieho procesu. Súťažné úlohy boli veľmi dobre vybrané a svojou úrovňou mierne prevyšovali úroveň bežnú na podujatiach tohto druhu. Nesporným prínosom bolo aj

pozvanie pomerne veľkého počtu hosťujúcich krajín (Bielorusko, Estónsko, Holandsko, Juhoslávia, Litva, Lotyšsko, Nemecko, Ukrajina a USA).

Budúci ročník Stredoeurópskej olympiády v informatike sa zaviazalo zorganizovať Chorvátsko. Dúfame, že tentokrát nebude potrebné hľadať náhradné riešenie v inej krajine, ako tomu bolo pred dvoma rokmi.

Bronislava Brejová, Tomáš Vinař

Zadania úloh 4. Stredoeurópskej olympiády v informatike

1. Jaskyňa (48 bodov)

V Bytelande sa nachádza veľa jaskýň. Všetky jaskyne v Bytelande majú tieto vlastnosti:

- Všetky siene a chodby jaskyne sú na tej istej úrovni.
- Žiadne dve chodby sa nekrižujú.
- Niektoré zo siení sa nachádzajú na vonkajšom okruhu. Tieto budeme nazývať *vonkajšie siene*.
- Všetky ostatné siene ležia vo vnútri vonkajšieho okruhu a budeme ich nazývať *vnútorné siene*.
- Jaskyňa má vchod, ktorý vedie do jednej z vonkajších jaskýň.
- Z každej siene vedú práve tri chodby do troch iných, navzájom rôznych siení. Ak je sieň vonkajšia, dve z jej chodieb vedú do dvoch vonkajších siení susediacich s ňou na vonkajšom okruhu. Tretia chodba vedie do niektorej vnútornej siene.
- Chodby spájajúce vonkajšie siene budeme nazývať *vonkajšie chodby*. Ostatné chodby budeme nazývať *vnútorné chodby*.
- Medzi každými dvoma sieňami sa dá prejsť tak, že použijeme iba vnútorné chodby. Existuje práve jedna takáto cesta, ak navyše predpokladáme, že môžeme navštíviť každú sieň nanajvýš raz.
- Nie cez všetky chodby je rovnako ťažké prejsť. Chodby sú rozdelené do dvoch skupín: ľahko a ťažko prechodné.

Rozhodlo sa, že všetky jaskyne budú prístupné pre verejnosť. Aby bol zabezpečený plynulý a bezpečný prechod cez jaskyne, návštevníci musia prechádzať jaskyne cestou, ktorá je vopred vyznačená a navštíviť každú sieň práve raz. Výnimkou z tohto pravidla je vstupná sieň, v ktorej každá prehliadka začína aj končí, túto sieň navštívia návštevníci práve dvakrát. Trasa prehliadky musí byť vhodná pre priemerného návštevníka, a teda musí obsahovať najmenší možný počet ťažko prechodných chodieb.

Úloha. Napíšte program, ktorý:

- načíta popis jaskyne z textového súboru CAV.IN;
- nájde trasu cez jaskyňu, ktorá začína aj končí vo vstupnej sieni tak, aby návštevníci videli každú sieň iba raz a ktorá obsahuje najmenší možný počet ťažkých chodieb;
- vypíše výsledok do textového súboru CAV.OUT.

Vstup. Na prvom riadku textového súboru CAV.IN sa nachádzajú dve celé čísla n a k oddelené jedinou medzerou. Celé číslo n ($3 < n \leq 500$) určuje počet všetkých siení v jaskyni a k ($k \leq 3$) je počet všetkých vonkajších siení jaskyne. Siene sú očíslované

od 1 po n . Sieň číslo 1 je vstupná, siene $1, 2, \dots, k$ sú vonkajšie siene. Nemusia sa však na vonkajšom okruhu vyskytovať v tomto poradí.

Nasledujúcich $\frac{3n}{2}$ riadkov obsahuje popisy jednotlivých chodieb. Každý popis pozostáva z troch celých čísel a, b, c , každé dve z nich sú oddelené jedinou medzerou. Čísla a a b označujú siene na koncoch príslušnej chodby; číslo c je 0 alebo 1, kde 0 znamená, že chodba je ľahko a 1, že chodba je ťažko prechodná.

Výstup. Váš program má na prvý riadok textového súboru `CAV.OUT` vypísať postupnosť n celých čísel oddelených medzerami (jedna medzera medzi každými dvoma číslami). Postupnosť musí začínať číslom 1 (vstupná sieň). Nasledujúcich $n - 1$ celých čísel budú po sebe nasledujúce siene na nájdenej trase.

2. Hexadecimálne čísla (30 bodov)

Základom hexadecimálnej sústavy je číslo 16. Na zápis čísel v tejto sústave sa používa 16 číslic $0, 1, \dots, 9, A, B, C, D, E, F$. Veľké písmená A, \dots, F označujú cifry $10, \dots, 15$ v tomto poradí. Napríklad hexadecimálne číslo $CF2$ má hodnotu $12 \cdot 16^2 + 15 \cdot 16 + 2 = 3314$ v desiatkovej sústave. Nech X je množina všetkých kladných celých čísel, ktorých hexadecimálny zápis má najviac osem cifier a žiadna cifra sa v tomto zápise neopakuje. Najvýznamnejšia cifra v takomto zápise nie je nula. Najväčším prvkom množiny X je číslo s hexadecimálnym zápisom $FEDCBA98$, druhé najväčšie je číslo $FEDCBA97$, tretie je $FEDCBA96$ a tak ďalej.

Úloha. Napíšte program, ktorý:

- načíta kladné číslo n z textového súboru `HEX.IN`, n nepresiahne počet prvkov X ;
- nájde n -tý najväčší prvok množiny X ;
- zapíše výsledok do textového súboru `HEX.OUT`.

Vstup. Prvý riadok súboru obsahuje celé číslo n v desiatkovej sústave.

Výstup. Váš program má do prvého riadku textového súboru `HEX.OUT` vypísať n -tý najväčší prvok množiny X v hexadecimálnom zápise.

3. Celočíselné intervaly (22 bodov)

Celočíselný interval $[a, b]$, $a < b$, je množina všetkých za sebou idúcich celých čísel od a po b vrátane.

Úloha. Napíšte program, ktorý:

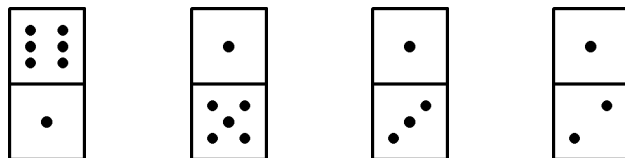
- načíta počet intervalov a ich popisy z textového súboru `INT.IN`;
- nájde najmenší počet prvkov v množine obsahujúcej aspoň dve rôzne celé čísla z každého intervalu;
- zapíše výsledok do textového súboru `INT.OUT`.

Vstup. Prvý riadok textového súboru `INT.IN` obsahuje počet intervalov n , $1 \leq n \leq 10\,000$. Každý z nasledujúcich n riadkov obsahuje dve celé čísla a, b oddelené jedinou medzerou, $0 \leq a < b \leq 10\,000$, určujúce začiatok a koniec intervalu.

Výstup. Váš program má zapísať jedno celé číslo do prvého riadku textového súboru `INT.OUT`; toto číslo má určovať minimálny počet prvkov množiny obsahujúcej aspoň dve rozličné celé čísla z každého intervalu.

4. O dominách (40 bodov)

Domino je tehlička veľkosti palca, ktorej vrchná časť je rozdelená na dva štvorce. Každý z týchto štvorcov je buď prázdny, alebo obsahuje jednu až šesť bodiek. Na stole je niekoľko domín položených vedľa seba:



Počet bodiek vo vrchnom riadku je $6+1+1+1=9$ a počet bodiek v spodnom riadku je $1+5+3+2=11$. Rozdiel medzi vrchným a spodným riadkom je 2. Vo všeobecnosti *rozdiel riadkov* je absolútna hodnota rozdielu medzi príslušnými dvoma súčtami. Každé domino môžeme otočiť o 180 stupňov tak, aby vrchná strana ostala otočená hore.

Aký je najmenší počet otočení potrebných na dosiahnutie minimálneho *rozdielu riadkov*? (V prípade na obrázku stačí otočiť posledné domino na dosiahnutie rozdielu 0. V tomto prípade je teda odpoveď 1.)

Úloha. Napíšte program, ktorý:

- načíta počet domín a ich popisy z textového súboru DOM.IN;
- vypočíta najmenší počet otočení potrebných na dosiahnutie minimálneho rozdielu riadkov;
- zapíše výsledok do textového súboru DOM.OUT.

Vstup. Prvý riadok textového súboru DOM.IN obsahuje celé číslo n , $1 \leq n \leq 1000$. Toto číslo udáva počet domín položených na stole.

Každý z nasledujúcich n riadkov obsahuje dve celé čísla a, b oddelené jedinou medzerou, $0 \leq a, b \leq 6$. Čísla a a b uvedené v riadku $i+1$ vstupného súboru, $1 \leq i \leq n$, určujú počty bodiek vo vrchnom a spodnom štvorci i -teho domina na stole.

Výstup. Váš program má do prvého riadku textového súboru DOM.OUT vypísať jediné celé číslo. Toto číslo má určovať najmenší počet otočení potrebných na dosiahnutie minimálneho rozdielu riadkov.

5. Prechod cez rieku (30 bodov)

Občania krajiny Byteland milujú všetky športy, v ktorých je logické uvažovanie rovnako dôležité ako fyzické schopnosti. Jedným z týchto športov je prechod cez rieku Hex – najširšiu rieku v Bytelande. V rieke sa nachádza n kolov očíslovaných $1, 2, \dots, n$ (zľava doprava) v rade tiahnúcim sa z jedného brehu na druhý. Vzďialenosti medzi týmito kolmi, ako aj vzdialenosť kolu 1 a ľavého brehu, alebo kolu n a pravého brehu sú rovnaké. Úlohou občanov je prejsť z jedného brehu rieky na druhý tak, že z *ľavého brehu* preskočia na niektorý *kôl*, potom prípadne na ďalší atď., až nakoniec na *pravý breh* rieky.

V čase 0 stojí občan na ľavom brehu rieky Hex. Jeho cieľom je dosiahnuť pravý breh rieky najrýchlejšie, ako je to možné. V každom okamihu je každý kôl buď *nad vodou* alebo *pod vodou* a občan stojí buď na niektorom kole alebo na jednom z brehov rieky. Občan môže stáť na kole iba ak je kôl nad vodou, takýto kôl sa nazýva *dostupný*.

Každý kôl je v čase 0 pod vodou, potom zostane a časových jednotiek nad vodou, b časových jednotiek pod vodou, a časových jednotiek nad vodou, b časových jednotiek pod vodou atď. Konštanty a a b sú definované zvlášť pre každý kôl. Napríklad kôl, pre ktorý $a = 2$ a $b = 3$, bude pod vodou v čase 0, nad vodou v čase 1 a 2, pod vodou v čase 3, 4, 5 atď.

V čase $t + 1$ si občan môže vybrať ľubovoľný dostupný kôl alebo breh vo vzdialenosti najviac 5 kolov od jeho pozície v čase t , alebo môže zostať na brehu alebo na pôvodnom kole (ak bude dostupný v čase $t + 1$). Napríklad z kolu 5 je možné dosiahnuť ľubovoľný z kolov 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 alebo ľavý breh.

Úloha. Napíšte program, ktorý:

- načíta počet blokov dát z textového súboru RIV.IN (každý blok obsahuje úplnú sadu dát pre úlohu);
- pre každý blok
 - načíta počet kolov a popis ich správania;
 - vypočíta najkratší čas, za ktorý sa môže občan dostať na pravý breh, ak je pravý breh dosiahnuteľný;
 - zapíše výsledok do textového súboru RIV.OUT.

Vstup. Prvý riadok vstupného súboru RIV.IN obsahuje počet blokov dát x , $1 \leq x \leq 5$. Nasledujúce riadky súboru tvoria x blokov. Prvý blok začína na druhom riadku vstupného súboru, každý ďalší blok začína priamo po predchádzajúcom bloku.

Prvý riadok každého bloku obsahuje počet kolov n , $5 < n \leq 1000$.

Každý z nasledujúcich n riadkov bloku obsahuje dve celé čísla a, b oddelené jedinou medzerou, $1 \leq a, b \leq 5$. Čísla v $i + 1$ -vom riadku bloku ($1 \leq i \leq n$) popisujú správanie kolu i .

Výstup. Pre k -ty blok, $1 \leq k \leq x$, zapíšte do k -teho riadku textového súboru RIV.OUT celé číslo, určujúce najkratší čas, za ktorý sa môže občan dostať na pravý breh alebo slovo NO, ak nie je možné opísaným spôsobom pravý breh dosiahnuť.

6. Strelecká súťaž (30 bodov)

Vitajte v Bytelande na výročnej streleckej súťaži. Každý súťažiaci bude strieľať na cieľ tvorený obdĺžnikovou mriežkou. Cieľ pozostáva z $r \times c$ štvorcov umiestnených do r riadkov a c stĺpcov. Každý štvorec je ofarbený bielou alebo čiernou. V každom stĺpci sú práve dva biele štvorce a $r - 2$ čiernych štvorcov. Riadky sú zhora dole označené číslami $1, \dots, r$ a stĺpce sú zľava doprava označené číslami $1, \dots, c$. Každý strelec má c výstrelův.

Séria c výstrelův je *korektná*, ak v každom stĺpci je zasiahnutý práve jeden biely štvorec a neexistuje riadok, v ktorom by nebol zasiahnutý žiadny biely štvorec. Pomôžte strelcovi nájsť korektnú sériu výstrelův, ak nejaká existuje.

Úloha. Napíšte program, ktorý:

- načíta z textového súboru SH0.IN počet blokov dát (každý blok obsahuje úplnú sadu dát pre úlohu);
- pre každý blok
 - načíta rozmery cieľa a rozloženie bielych štvorcov;

- zistí, či existuje korektná séria výstrelov a ak áno, nájde jednu z nich;
- zapíše výsledok do textového súboru SH0.OUT.

Vstup. Prvý riadok vstupného súboru SH0.IN obsahuje počet blokov dát x , $1 \leq x \leq 5$. Nasledujúce riadky súboru tvoria x blokov. Prvý blok začína na druhom riadku vstupného súboru, každý ďalší blok začína priamo za predchádzajúcim.

Prvý riadok každého bloku obsahuje dve celé čísla r a c oddelené jedinou medzerou, $2 \leq r \leq c \leq 1000$. Tieto čísla určujú počet riadkov a stĺpcov (v tomto poradí). Každý z nasledujúcich c riadkov v bloku obsahuje dve celé čísla oddelené jedinou medzerou. Čísla v $(i + 1)$ -vom riadku bloku ($1 \leq i \leq c$) označujú riadky obsahujúce biele štvorce v i -tom stĺpci.

Výstup. Pre i -ty blok ($1 \leq i \leq x$) má váš program vypísať do i -teho riadku súboru SH0.OUT buď postupnosť c čísel riadkov (susedné čísla sú oddelené jednou medzerou) tvoriacich korektnú sériu výstrelov na biele štvorce v stĺpcoch $1, 2, \dots, c$, alebo slovo NO, ak taká séria neexistuje.

9. Medzinárodná olympiáda v informatike

V dňoch 30. novembra až 7. decembra 1997 sa konala v Kapskom meste (Juhoafrická republika) 9. Medzinárodná olympiáda v informatike (MOI). Súťaže sa zúčastnilo 229 súťažiacich zo 61 krajín sveta.

Slovenské družstvo bolo zostavené na základe výsledkov celoštátneho kola Matematickej olympiády kategórie P a výberového sústredenia, ktoré sa konalo v dňoch 8. – 14. júna 1997 na Matematicko-fyzikálnej fakulte Univerzity Komenského v Bratislave.

Výsledky výberového sústredenia:

1.	<i>Miroslav Dudík</i>	402,5	7.	<i>Ján Rusz</i>	277,3
2.	<i>Stanislav Funiak</i>	400,6	8.	<i>Peter Vasil</i>	253,2
3.	<i>Richard Kráľovič</i>	343,1	9.	<i>Vladimír Koutný</i>	251,4
4.	<i>Martin Hajduch</i>	341,0	10.	<i>Dávid Pál</i>	229,4
5.	<i>Vladimír Marko</i>	298,5	11.	<i>Martin Vašíček</i>	228,0
6.	<i>Ján Svoreň</i>	297,1	12.	<i>Roland Bott</i>	195,5

Slovensko teda na súťaži reprezentovali *Miroslav Dudík* z Gymnázia v Trebišove, *Stanislav Funiak* z Gymnázia v Sučanoch, *Martin Hajduch* z Gymnázia v Považskej Bystrici a *Richard Kráľovič* z Gymnázia J. Hronca v Bratislave. Vedúcou delegácie bola *RNDr. Gabriela Andrejková* z Prírodovedeckej fakulty Univerzity P. J. Šafárika v Košiciach, zástupcom vedúceho bol *Tomáš Vinař* z Matematicko-fyzikálnej fakulty Univerzity Komenského v Bratislave.

Samotná súťaž bola rozdelená do dvoch súťažných dní, pričom každý deň súťažiaci riešili 3 úlohy v čistom čase 5 hodín. Oproti minulým ročníkom MOI boli súťažné príklady odlišného charakteru. Všetky boli orientované na použitie heuristických metód, pričom sa hodnotilo skóre nájdeného riešenia oproti skóre najlepšieho riešenia nájdeného vzorovým programom. Na súťažiach MOI je takýto systém hodnotenia pomerne neobvyklý.

Zadania úloh neboli veľmi dobre pripravené. Na zasadnutiach vedení družstiev pred jednotlivými súťažnými dňami museli byť pôvodné formulácie výrazne upravované, aby ich bolo možné na súťaži použiť. Toto viedlo neskôr k technickým problémom pri vyhodnocovaní.

Aj napriek tejto situácii dosiahlo naše družstvo výborné výsledky, a to konkrétne:

Por.	Meno	Počet bodov	Medaila
15.	Miroslav Dudík	384	zlatá
26.	Stanislav Funiak	358	strieborná
39.	Martin Hajduch	335	strieborná
46.	Richard Kráľovič	311	strieborná

V neoficiálnom hodnotení krajín sa Slovensko umiestnilo na 2. mieste za Ruskom (1 zlatá a 3 strieborné medaily) a pred Čínou (1 zlatá, 2 strieborné a 1 bronzová

medaila), Poľskom (2 zlaté a 2 bronzové medaily) a Dánskom (2 zlaté a 1 strieborná medaila).

Okrem súťaže organizátori pripravili množstvo ďalšieho programu (exkurzie, večery venované kultúre Južnej Afriky a Portugalska (budúci organizátor MOI), deň medzinárodnej spolupráce (pod záštitou firmy Microsoft) a iné. Tento program spolu s otváracím a záverečným ceremoniálom bol veľmi dobre pripravený a vytvoril výborné podmienky pre nadviazanie medzinárodných kontaktov ako medzi súťažiacimi, tak aj medzi vedúcimi delegácií.

Počas MOI'97 sa tiež konalo zasadnutie výboru Stredoeurópskej olympiády v informatike (CEOI). Členovia výboru rozhodli, že ďalší ročník CEOI sa bude konať v Zadare (Chorvátsko) v dňoch 20. – 27. mája 1998.

MOI'98 sa bude konať v dňoch 5.–12. septembra 1998 v meste Setúbal v Portugalsku.

RNDr. Gabriela Andrejková, Tomáš Vinař

Zadania úloh 8. medzinárodnej olympiády v informatike

1. Prieskumný modul Marsu

Počas budúcej misie pristane na Marse modul, ktorý bude obsahovať niekoľko prieskumných vozidiel MEV (malé elektrické vozidlo). Modul vyloží všetky vozidlá na povrch Marsu na mieste pristátia. Z tohoto miesta sa potom vozidlá budú presúvať k vysielateľke, ktorá pristála neďaleko. Počas presunu k vysielateľke odoberajú vozidlá vzorky hornín. Každú vzorku horniny možno odobrať iba raz. Odoberie ju prvé vozidlo, ktoré k nej dorazí. Vzorku nemožno odobrať druhýkrát, avšak iné MEV môže pozíciou, kde sa vzorka nachádzala, prejsť.

Vozidlá nemôžu prechádzať nebezpečným terénom. Sú zostrojené tak, že sa môžu presúvať len na juh a na východ po ceste vedúcej políčkami mriežky od modulu k vysielateľke. Na jednom políčku mriežky môže byť v rovnakom čase aj viacero vozidiel MEV.

Upozornenie: Ak vozidlo MEV nemôže pokračovať v korektnom pohybe a nedosiahlo ešte vysielateľku, jeho vzorky sú definitívne stratené.

Povrch planéty medzi modulom a vysielateľkou je reprezentovaný pomocou mriežky $P \times Q$. Modul sa vždy nachádza na pozícii $(1, 1)$ a vysielateľka na pozícii (P, Q) . Typy terénu sú nasledovné: 0 – voľný terén, 1 – nebezpečný terén, 2 – terén so vzorkou.

	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td colspan="10" style="padding: 5px;">Modul</td> <td style="padding: 5px;">Riadok 1</td> </tr> <tr> <td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td> </tr> <tr> <td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">1</td><td style="padding: 5px;">1</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;"></td> </tr> <tr> <td style="padding: 5px;">...</td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;"></td><td style="padding: 5px;">...</td><td style="padding: 5px;">...</td> </tr> <tr> <td style="padding: 5px;">0</td><td style="padding: 5px;">1</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">2</td><td style="padding: 5px;">0</td><td style="padding: 5px;">1</td><td style="padding: 5px;">1</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;"></td> </tr> <tr> <td style="padding: 5px;">0</td><td style="padding: 5px;">1</td><td style="padding: 5px;">0</td><td style="padding: 5px;">1</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">1</td><td style="padding: 5px;">1</td><td style="padding: 5px;">0</td><td style="padding: 5px;">0</td><td style="padding: 5px;">Riadok Q</td> </tr> <tr> <td colspan="10" style="padding: 5px;"></td> <td style="text-align: center; padding: 5px;">Vysielateľka</td> </tr> </table>	Modul										Riadok 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0		0	1	0	0	2	0	1	1	0	0		0	1	0	1	0	0	1	1	0	0	Riadok Q											Vysielateľka
Modul										Riadok 1																																																																				
0	0	0	0	0	0	0	0	0	0	0																																																																				
0	0	0	0	0	1	1	0	0	0																																																																					
...																																																																												
0	1	0	0	2	0	1	1	0	0																																																																					
0	1	0	1	0	0	1	1	0	0	Riadok Q																																																																				
										Vysielateľka																																																																				

Úloha. Nájdite popis pohybu vozidiel s maximálnym skóre. Skóre závisí od počtu zozbieraných a k vysielateľke prinesených vzoriek, ako aj od počtu vozidiel MEV, ktoré dorazili k vysielateľke.

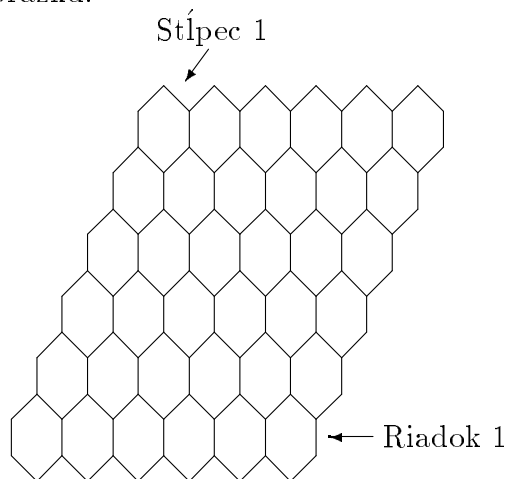
Vstup. Vstupný súbor je zadaný nasledovne: V prvom riadku sa nachádza celé číslo `PocetVozidiel` menšie ako 1000, označujúce počet MEV, ktoré boli dovezené modulom. Hodnota P v druhom a Q v treťom riadku vstupného súboru označujú veľkosť mriežky. Každý z nasledujúcich Q riadkov obsahuje P medzerami oddelených celých čísel z intervalu $[0, 2]$, reprezentujúcich typy terénu na jednom riadku mriežky povrchu, počínajúc riadkom 1 a končiac riadkom Q . Pritom P ani Q neprekročia 128.

Výstup. Postupnosť riadkov reprezentujúca pohyb vozidiel MEV k vysielateľke. Každý riadok obsahuje číslo vozidla a číslicu 0 resp. 1, kde 0 znamená posun na juh a 1 posun na východ.

Výpočet skóre. Skóre závisí od počtu zozbieraných vzoriek prinesených k vysielateľke a od počtu vozidiel MEV, ktoré k vysielateľke dorazili. Skóre = (počet zozbieraných a k vysielateľke donesených vzoriek + počet vozidiel MEV, ktoré dorazili k vysielateľke – počet vozidiel, ktoré nedorazili) vyjadrené percentuálne vzhľadom na maximálne možné skóre pre daný vstup.

2. Hra Hex

Hex je strategická hra pre dvoch hráčov, ktorá sa hrá na hracom pláne rovnobežníkového tvaru $N \times N$, pozostávajúceho zo šesťuholníkových políčok (*hexovnica*). Tvar pre $N = 6$ je uvedený na obrázku:



- Hráčmi sú váš program a vyhodnocovacia knižnica.
- Váš program začína.
- Hráči sa striedajú v kladení hracích kameňov (*hexovníčkov*) na hexovnicu.
- Hexovníček môže byť umiestnený na ľubovoľné voľné políčko na hexovnici.
- Dve políčka hexovnice sú susedné, práve ak majú spoločnú hranu.

- Dva hexovníčky (toho istého hráča), ktoré sa nachádzajú na susedných políčkach hexovnice sú *spojené*.
- Spojitosť je tranzitívna (a komutatívna): ak je hexovníček hex_1 spojený s hexovníčkom hex_2 a hex_2 je spojený s hexovníčkom hex_3 , potom hex_3 je spojený s hex_1 a hex_1 je spojený s hex_3 .

Úloha.

- Vytvorte program, ktorý hrá hru Hex.
- Cieľom prvého hráča (váš program) je spojiť niektorý z vašich hexovníčkov v stĺpci 1 s niektorým z vašich hexovníčkov v stĺpci N .
- Cieľom druhého hráča (vyhodnocovací program) je spojiť jeden z jeho hexovníčkov v riadku 1 s niektorým z jeho hexovníčkov v riadku N .
- Ak váš program hrá optimálne, vždy vyhrá.

Vstup a výstup. Váš program **nesmie** čítať a zapisovať žiadne súbory. Váš program **nesmie** očakávať vstup z klávesnice a **nesmie** vypisovať na obrazovku. Program získa všetky vstupné údaje pomocou funkcií v knižnici HexLib. Knižnica bude vytvárať výstupný súbor HEX.OUT; tento výstup ignorujte.

Na začiatku vášmu programu bude zadaná hexovnica, na ktorej môže byť už uložených niekoľko hexovníčkov, reprezentujúcich stav hry taký, že prvý hráč môže stále vyhrať. Váš program na zistenie stavu hexovnice musí použiť funkcie GetMax a LookAtBoard.

Na začiatku patrí vášmu programu aj vyhodnocovacej knižnici rovnaký počet hexovníčkov.

Obmedzenie.

- Veľkosť hexovnice bude vždy od 1 po 20.
- Váš program musí ukončiť hru do 200 ťahov. Celá hra musí byť ukončená do 40 sekúnd. Je zaručené, že vyhodnocovacia knižnica ukončí spracovanie všetkých ťahov spolu do 20 sekúnd.

Popis knižnice. K dispozícii máte knižnicu HexLib, ktorú musíte prilinkovať k vášmu programu.

Nasleduje popis funkcií v knižnici HexLib (v poradí Pascal, C/C++):

```
function LookAtBoard (row, column:integer):integer; int LookAtBoard(int
row; int column);
```

vráti:

- 1 ak $row < 1$ alebo $row > N$ alebo $column < 1$ alebo $column > N$,
- 0 ak sa na políčku nenachádza žiadny hexovníček,
- 1 ak sa na políčku nachádza váš hexovníček,
- 2 ak sa na políčku nachádza hexovníček patriaci knižnici.

procedure PutHex(*row*, *column*:integer); void PutHex(int *row*, int *column*);

umiestni váš hexovníček na políčko určené *row* a *column*, ak toto políčko nie je obsadené.

function GameIsOver:integer; int GameIsOver(void);

vráti jedno z nasledujúcich celých čísel:

0	ak hra nie je ukončená,
1	ak každá pozícia je obsadená,
2	ak váš program vyhral,
3	ak vyhrala knižnica.

procedure MakeLibMove; void MakeLibMove(void);

umožní knižnici vypočítať ďalší ťah a umiestni jej hexovníček na hexovnicu. Zmeny na hexovnici zistíte pomocou `LookAtBoard` a iných funkcií.

function GetRow:integer; int GetRow(void);

vráti číslo riadku, na ktorý bol knižnicou umiestnený posledný hexovníček alebo -1 , ak doteraz nebol umiestnený žiadny hexovníček. Táto funkcia vždy vráti tú istú hodnotu, až kým váš program nezavolá `MakeLibMove` znovu.

function GetColumn:integer; int GetColumn(void);

vráti číslo stĺpca, na ktorý bol knižnicou umiestnený posledný hexovníček alebo -1 , ak doteraz nebol umiestnený žiadny hexovníček. Táto funkcia vždy vráti tú istú hodnotu, až kým váš program nezavolá `MakeLibMove` znovu.

function GetMax:integer; int GetMax(void);

vráti veľkosť hexovnice N .

3. Jedovatý iShongololo

iShongololo je v reči Zulu meno pre zvláštnu stonožku. iShongololo je dlhé, lesklé a čierne článkované zvieratko s mnohými nožičkami. Živí sa „ovocím“, ktoré pre účely tohto príkladu budeme považovať za kváder s celočíselnými rozmermi L (dĺžka), W (šírka), H (výška).

Úloha. Vytvorte program, ktorý určí maximálny počet blokov zjedených zvieratkom iShongololo bez porušenia pravidiel. Program musí určiť činnosti tak, ako ich iShongololo bude postupne vykonávať pri požívaní ovocia.

iShongololo sa na začiatku nachádza mimo ovocia. Prvý blok, ktorý iShongololo musí zjesť, je $1, 1, 1$, a potom sa musí posunúť dovnútra tohoto bloku. Zastaví sa, keď už nemožno žiadne ďalšie bloky zjesť podľa pravidiel a nemôže sa viac posunúť.

Obmedzenia.

- iShongololo zaberá práve jeden prázdny blok.

- iShongololo môže zjesť naraz iba jeden kompletný blok.
- iShongololo sa nemôže dvakrát posunúť na to isté miesto (t.j. posunúť sa naspäť alebo križovať svoju cestu).
- iShongololo sa nemôže posunúť do nezjedeného bloku, ani mimo ovocia.
- iShongololo sa môže posúvať do alebo zjesť iba tie bloky, s ktorými má blok, v ktorom sa nachádza, spoločnú stenu. Navyše môže zjesť iba tie bloky, ktoré nemajú žiadne iné steny susedné s už zjedenými (prázdny)mi blokmi.

Vstup. Váš program dostane na vstup tri celé čísla, ktoré označujú dĺžku L , šírku W a výšku H kvádra. Každé z čísel L , W , H sa nachádza na samostatnom riadku. Všetky tri čísla budú od 1 do 32 (vrátane).

Výstup. Výstup pozostáva z riadkov, ktoré začínajú písmenom E (zjedz – eat) alebo M (posuň sa – move). Za týmto písmenom nasledujú tri celé čísla, ktoré reprezentujú zjedený blok alebo blok, do ktorého sa iShongololo presunul.

Hodnotenie. Celkové skóre je percentuálnym vyjadrením počtu zjedených blokov vzhľadom k autormi úlohy nájdenému najvyššiemu počtu zjedených blokov.

4. Rozpoznávanie znakov

Úloha. Napíšte program na rozpoznanie postupnosti jedného alebo viacerých znakov v obraze, ktorý sa nachádza v súbore IMAGE.DAT s použitím fontov, ktoré sa nachádzajú v súbore FONT.DAT.

Každý ideálny obraz znaku má 20 riadkov po 20 číslic. Každá číslica je 0 alebo 1 (pozri príklad obrazu znaku v súbore FONT.DAT alebo IMAGE.DAT).

Súbor FONT.DAT obsahuje reprezentácie 27 ideálnych obrazov znakov v tomto poradí:

□abcdefghijklmnopqrstuvwxyz,

kde □ reprezentuje medzeru.

Súbor IMAGE.DAT obsahuje jeden alebo niekoľko potenciálne pokazených obrazov znakov. Obraz znaku môže byť pokazený nasledujúcimi spôsobmi:

- najviac jeden riadok môže byť zdvojený (a kópia sa nachádza bezprostredne za originálom)
- najviac jeden riadok môže byť vynechaný
- niektoré 0 môžu byť zmenené na 1
- niektoré 1 môžu byť zmenené na 0. Žiadny obraz znaku nemá súčasne zdvojený aj vynechaný riadok. Vo vstupných údajoch pre vyhodnocovanie v žiadnom obraze znaku nie je zmenených viac než 30% núl a jedničiek.
- v prípade zdvojenia riadku, jeden alebo oba výsledné riadky môžu byť pokazené, a toto pokazenie môže byť pre oba riadky rôzne.

Pokazený obraz znaku rozpoznať tak, že vyberiete obraz takého znaku z fontu, ktorý vyžaduje najmenší celkový počet zmien jedničiek na nuly a naopak, aby ste získali

Kontajnery prichádzajú do skladu každú hodinu a zostávajú tam kladný celočíselný počet hodín. Pri uschovaní kontajnera je v jeho dokumentácii obsiahnutý predpokladaný čas, kedy bude vyzdvihnutý. Prvý kontajner je uschovaný v čase 1. Skutočný čas, v ktorom bude kontajner vyzdvihnutý, sa líši od predpokladaného času nanajviš o 5 hodín. V tejto úlohe je čas meraný v hodinách a nepresiahne hodnotu 150.

Sklad má tvar kvádra s dĺžkou X , šírkou Y a výškou Z . Každý kontajner je kocka $1 \times 1 \times 1$. Kontajnery sú ukladané na iné kontajnery alebo na podlahu žeriavom. Žeriav sa môže pohybovať len nad daným úložným priestorom. Uschováva doň a vyzdvihuje z neho kontajnery a niekedy ich preusporiada vo vnútri úložného priestoru. Môže manipulovať iba s kontajnermi, ktoré sú na vrchu ktoréhokoľvek stĺpca kontajnerov.

Presúvanie kontajnera z jednej pozície na inú je vždy považované za jeden presun žeriava. Všetky presuny žeriava sa vykonávajú medzi uloženiami resp. vyzdvihnutiami kontajnerov a vykonávajú sa okamžite.

Úloha. Napíšte program s dobrou stratégiou na uschovávanie, skladovanie a vyzdvihovanie kontajnerov. Dobrá stratégia je taká, ktorá minimalizuje celkový počet posunutí, ktoré žeriav vykoná.

Keď sa sklad naplní, váš program musí odmietnuť uloženie ďalších kontajnerov. V prípade, že sklad je skoro plný, sa váš program môže stať menej efektívnym, alebo môže byť vôbec neschopný pokračovať. Môže teda odmietnuť uloženie ďalšieho kontajnera kedykoľvek.

Vstup. váš program bude spolupracovať so simulačným modulom, ktorý bude poskytovať údaje, a do ktorého bude váš program oznamovať činnosti a posielat správy. Do vášho programu musíte prilinkovať knižnicu `StackLib`. Počas ladenia vášho programu bude knižnica vracat hodnoty pre jeden malý konkrétny vstup.

Sklad je pri spustení vášho programu prázdny a jeho rozmery X , Y a Z neprekročia hodnotu 32.

Váš program môže hocikedy volat nasledujúce funkcie:

int GetX(); function GetX:integer; – vráti dĺžku úložného priestoru (celé číslo).

int GetY(); function GetY:integer; – vráti šírku úložného priestoru (celé číslo).

int GetZ(); function GetZ:integer; – vráti výšku úložného priestoru (celé číslo).

Každý kontajner je jednoznačne identifikovaný kladným celým číslom. Nasledujúce funkcie poskytujú informácie o postupnosti činností (uschovanie a vyzdvihnutie kontajnera). Požiadavky na uschovanie prichádzajú vždy presne v celú hodinu a požiadavky na vyzdvihnutie vždy medzi dvoma celými hodinami. **Pre účely predpokladaného času uschovania každá požiadavka na uschovanie znamená uplynutie jednej hodiny času.**

int GetNextContainer(); function GetNextContainer:integer;

Vráti číslo kontajnera, ktorý má byť najbližšie uschovaný alebo vyzdvihnutý. Ak nie sú žiadne ďalšie požiadavky na vyzdvihnutie alebo uschovanie kontajnera, vráti funkcia hodnotu 0, čo znamená, že váš program má ukončiť činnosť, a to aj v prípade, že nejaké kontajnery ešte zostali v sklade.

int GetNextAction(); function GetNextAction:integer;

Vráti celé číslo, ktoré reprezentuje najbližšiu požadovanú činnosť: 1 – uschovať nový kontajner, 2 – vyzdvihnúť kontajner.

int GetNextStorageTime(); function GetNextStorageTime:integer;

Vráti v hodinách (absolútne od začiatku behu programu) predpokladaný čas vyzdvihnutia kontajnera. Táto hodnota je poskytovaná pre plánovanie vo vašom programe; skutočná požiadavka na vyzdvihnutie môže prísť o niečo skôr alebo neskôr (nanajvýš však o 5 hodín). Hodnota funkcie `GetNextStorageTime` má význam iba v prípade, že funkcia `GetNextAction` vráti hodnotu 1.

Nezáleží na poradí volania vyššie uvedených troch funkcií. Po sebe nasledujúce volania `GetNextContainer`, `GetNextAction` a `GetNextStorageTime` vrátia vždy informáciu o tom istom kontajneri, až kým tento nebude odmietnutý, uschovaný alebo vyzdvihnutý (od tohto momentu vyššie uvedené funkcie budú vracaať informáciu o ďalšom kontajneri).

Výstup. Keď váš program získa potrebnú informáciu o ďalšom kontajneri, použite nasledujúce funkcie na manipuláciu s úložným priestorom:

int MoveContainer(int x1, int y1, int x2, int y2);

function MoveContainer(x1,y1,x2,y2:integer):integer;

Vezme kontajner, ktorý sa nachádza na vrchu stĺpika kontajnerov na pozícii $(x1, y1)$ a položí ho na vrch stĺpika kontajnerov na pozícii $(x2, y2)$. Vráti 1, ak operáciu možno vykonať, 0, ak operáciu nemožno vykonať.

void RefuseContainer(); procedure RefuseContainer();

Odmietne uschovať kontajner.

void StoreArrivingContainer(int x, int y);

procedure StoreArivingContainer(x,y:integer);

Uloží kontajner, ktorý je potrebné uschovať, na vrchol stĺpika kontajnerov na pozícii (x, y) .

void RemoveContainer(int x, int y);

procedure RemoveContainer(x,y:integer);

Vyzdvihne kontajner z vrcholu stĺpika kontajnerov na pozícii (x, y) zo skladu.

Ak váš program nemôže vykonať požadovanú činnosť, musí skončiť. Nesprávne kroky sú knižnicou ignorované a nemajú žiadny efekt na stav simulácie ani na hodnotenie.

Od vášho programu sa nevyžaduje žiadny výstupný súbor. Knižnica, s ktorou váš program spolupracuje, bude vypisovať log súbor činností. Tento súbor bude používaný na vyhodnocovanie.

Časovanie. Váš program dostane informáciu o ďalšej požiadavke na uschovanie alebo vyzdvihnutie kontajnera. Potom, ak je to potrebné, môže pomocou žeriava presúvať a postupne uschovávať, vyzdvihovať, alebo odmietat kontajnery.

Hodnotenie. Program bude testovaný pomocou niekoľkých vstupov a pre každý vstup bude hodnotená jeho výkonnosť oproti nám známemu najefektívnejšiemu riešeniu použitím nasledujúcich ukazovateľov:

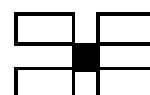
- Celkový počet presunov žeriava vo vašom programe.
- Penalizácia 5 presunov za každý odmietnutý kontajner.
- Penalizácia 5 presunov za každý neuschovaný a nevyzdvihnutý kontajner (ak program skončí normálne pred tým, než sú všetky operácie vykonané).
- Celkové skóre bude vypočítané vzhľadom k najlepšiemu známemu riešeniu.
- Ak program vykoná viac ako dvojnásobok nutných operácií, je hodnotený 0 bodmi.

6. Popisovanie máp

Predstavte si, že ste asistentom kartografa a dostali ste za úlohu zapísať mená miest do novej mapy.

Mapa je reprezentovaná mriežkou $1\,000 \times 1\,000$ políčok. Každé mesto zaberá na mape práve jedno políčko. Mená miest majú byť umiestnené na mape do obdĺžnikov, ktoré pozostávajú z políčok mriežky. Tieto obdĺžniky nazvime *popiskami*.

Mesto so štyrmi možnými umiestneniami popisky:



Umiestnenie popisiek musí spĺňať nasledujúce obmedzenia:

- popiska mesta sa musí nachádzať v jednej zo štyroch pozícií vzhľadom na mesto tak, ako je to nakreslené na obrázku vyššie;
- popisiky sa nesmú prekrývať;
- popisiky nesmú prekrývať mestá;
- popisiky sa musia celé nachádzať na mape.

Každá popiska obsahuje všetky písmená z názvu mesta a jednu medzeru. Pre každé mesto bude daná šírka a výška písmen jeho názvu; medzera má takú istú veľkosť ako písmeno.

	L	a	n	g	a	□											
■							■										
								P	a	a	r	l	□				
□	C	e	r	e	s												

□ reprezentuje medzeru
■ reprezentuje pozíciu mesta

Stĺpce sú očíslované zľava doprava, pričom najľavejší stĺpec má číslo 0. Riadky sú očíslované zdola nahor, pričom najspodnejší riadok má číslo 0. Na obrázku je zobrazená spodná ľavá časť mapy pre mestá Langa (na pozícii $(0, 3)$), Ceres (na pozícii $(6, 1)$) a Parl (na pozícii $(7, 3)$). Všetky popisky sú správne umiestnené, aj keď toto umiestnenie nie je jediné možné.

Úloha. Váš program načíta rozmiestnenie miest na mape, za ktorým nasledujú rozmery písmen a meno mesta. Program má rozmiestniť na mape najväčší možný počet popisiek miest, pričom neporuší vyššie uvedené obmedzenia. Do výstupného súboru program vypíše rozmiestnenie popisiek, ktoré boli na mapu umiestnené.

Vstup. Vstupný súbor začína riadkom obsahujúcim celé číslo N , ktoré vyjadruje počet miest na mape. Pre každé mesto sa ďalej v súbore nachádza riadok obsahujúci číselné údaje popisujúce

horizontálnu súradnicu X polohy mesta, vertikálnu súradnicu Y polohy mesta, celočíselnú šírku W písmen v mene mesta, celočíselnú výšku H písmen v mene mesta a samotné meno mesta, ktoré je vždy jednoslovné.

Jednotlivé údaje sú oddelované medzerou, počet miest je nanejvýš 1 000 a žiadne meno mesta nemá viac než 200 znakov.

Výstup. Váš program vytvorí výstupný súbor s N riadkami. Každý riadok bude obsahovať horizontálnu a vertikálnu súradnicu (v tomto poradí) polohy ľavého horného políčka popisky mesta (tieto čísla budú oddelené medzerou). Ak váš program neumiestnil popisku pre mesto, na výstup zapíše $-1 -1$.

Riadky musia byť vo výstupnom súbore v tom istom poradí, v akom sa mestá nachádzali vo vstupnom súbore.

Hodnotenie. Hodnotenie je percentuálnym vyjadrením počtu vašim programom umiestnených popisiek vzhľadom k riešeniu organizátorov. Ak niektorá popiska poruší obmedzenia, program získava 0 bodov, rovnako, ako v prípade, ak umiestnenie popisky nesúhlasí so skutočným menom príslušného mesta.

Korešpondenčný seminár SK MO

V 46. ročníku matematickej olympiády SK MO prebiehal pre najúspešnejších olympionikov predchádzajúceho ročníka MO zo Slovenska korešpondenčný seminár SK MO. Po minuloročnej zmene názvu seminára z KS ÚV MO na KS ÚK MO sa jeho názov zmenil na terajší KS SK MO. Tento korešpondenčný seminár vznikol už v 24. ročníku MO preto, aby bolo umožnené venovať individuálnu starostlivosť aj tým študentom, ktorí nenavštevovali triedy so zameraním na matematiku. V súčasnosti, pretože existuje veľké množstvo iných matematických korešpondenčných seminárov (napríklad krajských, ktorým je venovaná samostatná kapitola), a pretože počet škôl so zameraním na matematiku stúpol, seminár SK MO sa zameriava na zlepšenie prípravy všetkých študentov, ktorí preukázali svoje schopnosti v predchádzajúcich ročníkoch MO. Keďže úlohy tohoto seminára svojou náročnosťou prevyšujú akúkoľvek inú matematickú súťaž pre stredoškolákov, seminár sa stáva dôležitou súčasťou prípravy aj na medzinárodnú matematickú olympiádu. V 44. ročníku MO bol KS SK MO prvýkrát zorganizovaný samostatne na Slovensku. Pozostáva tradične z piatich sérií, v každej je zadaných sedem príkladov. Do riešenia sa v tomto ročníku zapojilo spolu 40 študentov zo všetkých krajov. Medzi desiatimi najúspešnejšími riešiteľmi boli všetci 6 členovia slovenskej delegácie na MMO, z nich piati skončili na prvých šiestich miestach.

Korešpondenčný seminár viedol *Richard Kollár* a opravovanie zabezpečovali študenti a pracovníci MFF UK (všetko bývalí olympionici).

Celkové poradie KS SK MO 1997/98

1. *Peter Kozák*, 2 Gymnázium Sučany, 69 bodov;
2. *Peter Novotný*, 2 Gymnázium Veľká Okružná, Žilina, 57, 5 bodu;
3. *Ján Špakula*, 3 Gymnázium Poštová, Košice, 56 bodov;
4. *Pavol Novotný*, 3 Gymnázium Veľká Okružná, Žilina, 52 bodov;
5. *Miroslav Dudík*, 4 Gymnázium Trebišov, 49, 5 bodu;
6. *Viera Růžičková*, 3 Gymnázium Veľká Okružná, 48, 5 bodu;
7. *Peter Svrček*, 3 Gymnázium Veľká Okružná, Žilina, 47, 5 bodu;
8. *Peter Bodík*, 3 Gymnázium Poštová Košice, 38 bodov.

Uvádzame všetky príklady tohto ročníka súťaže spolu s riešeniami, prevažne študentskými. Príklady boli vyberané z príkladov zo jury MMO a z národných olympiád, či iných súťaží týchto krajín: India, Bielorusko, Maďarsko, Izrael, Kanada, Japonsko.

Zadania súťažných úloh KS SK MO

PRVÁ SÉRIA

- 1.1** Množina M je takou podmnožinou množiny $\{1, 2, 3, \dots, 15\}$, že súčin žiadnych troch rôznych prvkov M nie je druhou mocninou prirodzeného čísla. Určte najväčší možný počet prvkov tejto množiny.
- 1.2** Nech a, b a c sú kladné reálne čísla, pre ktoré platí $abc = 1$. Dokážte, že platí

$$\frac{ab}{a^5 + b^5 + ab} + \frac{bc}{b^5 + c^5 + bc} + \frac{ca}{c^5 + a^5 + ca} \leq 1.$$

Kedy nastáva rovnosť?

- 1.3** Do štvorcov šachovnice 5×5 dvaja hráči striedavo vpisujú po jednom prirodzenom čísle. Prvý hráč vpisuje číslo 1, druhý číslo 0. Keď zaplnia celú šachovnicu, spočítajú súčet čísel v každom z deviatich možných štvorcov 3×3 . Označme A maximum týchto súčtov. Prvý hráč sa snaží v tejto hre dosiahnuť čo najväčšiu hodnotu A . Akú najväčšiu hodnotu A môže zaručiť, nezávisle na hre súpera?
- 1.4** Daná je funkcia $f, f: \mathbb{R} \rightarrow \langle -1, 1 \rangle$ taká, že pre každé $x \in \mathbb{R}$ platí:

$$f\left(x + \frac{13}{42}\right) + f(x) = f\left(x + \frac{1}{6}\right) + f\left(x + \frac{1}{7}\right).$$

Dokážte, že f je periodická funkcia.

- 1.5** Daný je lichobežník $ABCD$ s $BC \parallel AD$. Označme po rade M, P a Q stredy strán CD, MA a MB . Priesečník DP a CQ označme N . Dokážte, že bod N leží vnútri trojuholníka ABM .
- 1.6** Nech n, p, q sú prirodzené čísla, pre ktoré $n > p + q$. Nech x_0, x_1, \dots, x_n sú celé čísla spĺňajúce nasledujúce podmienky:
- $x_0 = x_n = 0$;
 - pre každé celé číslo $i, 1 \leq i \leq n$ je buď $x_i - x_{i-1} = p$, alebo $x_i - x_{i-1} = -q$. Ukážte, že existuje dvojica indexov (i, j) taká, že $i < j, (i, j) \neq (0, n)$ a $x_i = x_j$.
- 1.7** Nech $ABCDEF$ je konvexný šesťuholník taký, že AB je rovnobežné s DE , BC je rovnobežné s EF a CD je rovnobežné s AF . Označme R_A, R_C a R_E po rade polomery kružníc opísaných trojuholníkom FAB, BCD a DEF a nech p je obvod daného šesťuholníka. Dokážte, že platí $R_A + R_C + R_E \geq \frac{p}{2}$.

DRUHÁ SÉRIA

- 2.1** Pre každé prirodzené číslo n označme $S(n)$ najväčšie prirodzené číslo s nasledujúcou vlastnosťou: pre každé prirodzené číslo $k \leq S(n)$ môžeme číslo n^2 napísať ako súčet k štvorcov prirodzených čísel.
- a) Dokážte, že $S(n) \leq n^2 - 14$ pre $n \geq 4$.
- b) Nájdite prirodzené číslo n , pre ktoré $S(n) = n^2 - 14$.
- c) Dokážte, že existuje nekonečne veľa prirodzených čísel n , pre ktoré $S(n) = n^2 - 14$.
- 2.2** Dokážte, že spomedzi 101 obdĺžnikov so stranami kladnej celočíselnej dĺžky neprevyšujúcej 100 možno vybrať tri \mathcal{A} , \mathcal{B} a \mathcal{C} také, že \mathcal{A} sa vojde do \mathcal{B} a \mathcal{B} sa vojde do \mathcal{C} . (Obdĺžniky musia mať jednu stranu umiestnenú vodorovne.)
- 2.3** Daná je kružnica k , jej dotyčnica l a na l bod M . Nájdite geometrické miesto bodov P s nasledujúcou vlastnosťou: Na priamke l existujú body Q a R také, že M je stred QR a k kružnica vpísaná trojuholníku PQR .
- 2.4** Pre trojicu bodov P , Q a R ležiacich v tej istej rovine označme $m(PQR)$ minimum z dĺžok výšok trojuholníka PQR (ak sú P , Q a R kolineárne, $m(PQR) = 0$). V rovine sú dané sú tri body A , B a C . Dokážte, že pre ľubovoľný bod X tejto roviny platí:

$$m(ABC) \leq m(ABX) + m(AXC) + m(XBC).$$

- 2.5** Na nekonečnej šachovnici možno hrať nasledujúcu hru: Na začiatku je v niektorom zo štvorcov $n \times n$ umiestnených n^2 mydiel (v každom políčku jedno). V jednom ťahu môže mydlo preskočiť vo vodorovnom alebo zvislom smere cez mydlo na susednom políčku a premiestniť sa na voľné miesto hneď za týmto políčkom. Po tomto ťahu preskočené mydlo odložíme preč zo šachovnice. Nájdite tie n , pre ktoré sa hra môže skončiť len s jedným mydlom na šachovnici.
- 2.6** V trojuholníku ABC označme priesečníky osí uhlov β a γ s protíľahlými stranami po rade D a E . Určte uhly trojuholníka ABC , ak $|\sphericalangle BDE| = 24^\circ$ a $|\sphericalangle CED| = 18^\circ$.
- 2.7** Všimnime si, že číslo $12^2 = 144$ sa končí na dve štvorky. Určte dĺžku najdlhšieho bloku rovnakých nenulových číslic, na ktorý sa môže končiť štvorec prirodzeného čísla.

TRETIA SÉRIA

- 3.1** Nájdite všetky prirodzené čísla a a b pre ktoré platí

$$\left\lfloor \frac{a^2}{b} \right\rfloor + \left\lfloor \frac{b^2}{a} \right\rfloor = \left\lfloor \frac{a^2 + b^2}{ab} \right\rfloor + ab.$$

(Symbolom $\lfloor x \rfloor$ označujeme dolnú celú časť čísla x .)

- 3.2** V ostrouhlom trojuholníku ABC platí $|BC| > |CA|$. Označme O stred jeho opísanej kružnice, H jeho ortocentrum a F päťu jeho výšky CH . Ďalej označme P priesečník kolmice na priamku OF v bode F so stranou CA . Dokážte, že $|\sphericalangle FHP| = |\sphericalangle BAC|$.
- 3.3** Nech a_1, a_2, \dots, a_n , ($n \geq 3$), sú reálne čísla z intervalu $\langle 2, 3 \rangle$ a nech s je ich súčet. Dokážte, že

$$\frac{a_1^2 + a_2^2 - a_3^2}{a_1 + a_2 - a_3} + \frac{a_2^2 + a_3^2 - a_4^2}{a_2 + a_3 - a_4} + \dots + \frac{a_n^2 + a_1^2 - a_2^2}{a_n + a_1 - a_2} \leq 2s - 2n.$$

- 3.4** Strany dvoch obdĺžnikov sú po rade $\{a, b\}$ a $\{c, d\}$, kde $a < c \leq d < b$ a $ab < cd$. Dokážte, že prvý obdĺžnik možno umiestniť do druhého práve vtedy, keď

$$(b^2 - a^2)^2 \leq (bd - ac)^2 + (bc - ad)^2.$$

- 3.5** Na kružnici sú vyznačené štyri celé čísla. V každom kroku naraz vymeníme každé z čísel rozdielom tohto čísla a susedného čísla v smere hodinových ručičiek (teda čísla a, b, c a d budú nahradené číslami $a - b, b - c, c - d$ a $d - a$). Možno dosiahnuť, aby sme po 1997 takýchto krokoch dostali na kružnici štyri čísla a, b, c a d tak, že čísla $|bc - ad|$, $|ac - bd|$ a $|ab - cd|$ sú všetky prvočísla?
- 3.6** Nech $P(x)$ je polynóm s reálnymi koeficientami, $P(x) = ax^3 + bx^2 + cx + d$. Dokážte, že ak $|P(x)| \leq 1$ pre každé x , $|x| \leq 1$, tak

$$|a| + |b| + |c| + |d| \leq 7.$$

- 3.7** Nech k, m a n sú prirodzené čísla, pre ktoré platí $1 < n \leq m - 1 \leq k$. Určte najväčšiu mohutnosť podmnožiny \mathcal{S} množiny $\{1, 2, \dots, k\}$, v ktorej súčet žiadnych n rôznych prvkov nedáva súčet menší ako m .

ŠTVRTÁ SÉRIA

- 4.1** Dve muchy \mathcal{F} a \mathcal{M} lietajú vo vnútri miestnosti tvaru kocky s rozmermi $10 \text{ m} \times 10 \text{ m} \times 10 \text{ m}$. Na začiatku sú muchy \mathcal{F} a \mathcal{M} po rade v ľubovoľných bodoch A a B kocky. Po nejakom čase je \mathcal{F} v bode B a \mathcal{M} v bode A .
- a) Dokážte, že v nejakom čase bola vzdialenosť medzi \mathcal{F} a \mathcal{M} menšia ako 15 metrov.
- b) Zistite, či si mohli muchy vymeniť miesta tak, aby vzdialenosť medzi nimi bola po celý čas práve 14 metrov (pre ľubovoľnú začiatočnú polohu).
- 4.2** Polynómy $P(x)$ a $Q(x)$ s reálnymi koeficientami majúce aspoň jeden reálny koreň spĺňajú pre všetky reálne čísla x rovnosť

$$P(1 + x + Q(x)^2) = Q(1 + x + P(x)^2).$$

Dokážte, že $P(x) \equiv Q(x)$.

- 4.3** Nech O je stred vpísanej kružnice trojuholníka ABC a P, Q a R sú po rade ťažiská trojuholníkov BOC , AOC a AOB . Dokážte, že stred vpísanej kružnice trojuholníka PQR a ťažisko trojuholníka ABC sú totožné.
- 4.4** Do štvoruholníka $ABCD$ je vpísaná kružnica k . Body K, L, M a N sú po rade dotykové body tejto kružnice so stranami AB, BC, CD a DA štvoruholníka $ABCD$. Predĺženia strán DA a CB sa pretínajú v bode S a predĺženia strán BA a CD sa pretínajú v bode P . Dokážte, že ak body S, K a M ležia na jednej priamke, potom aj body P, N a L ležia na jednej priamke.
- 4.5** Daných je šesť úsečiek tak, že každá trojica z nich môže byť trojicou strán nejakého trojuholníka. Dvaja hráči hrajú s týmito úsečkami hru s nasledujúcimi pravidlami:
V každom ťahu si každý hráč vezme jednu úsečku. Začína prvý hráč a ďalej sa v ťahoch striedajú až kým nezoberie druhý hráč poslednú úsečku. Vyhráva ten z nich, ktorý môže z úsečiek poskladať trojuholník s väčším obsahom. Dokážte, že pre prvého hráča existuje neprehrávajúca stratégia.
- 4.6** Pre prirodzené čísla m a n platí $m(m+2) = 2n(n+2)$. Dokážte, že platí

$$\sum_{k=0}^n [k \cdot \sqrt{2}] = \sum_{k=0}^m \left\lfloor k \cdot \frac{\sqrt{2}}{2} \right\rfloor.$$

(Symbolom $[x]$ označuje dolnú celú časť čísla x .)

- 4.7** Zistite, či existuje prirodzené číslo n , $n > 1$, spĺňajúce nasledujúcu podmienku: Množina prirodzených čísel sa dá rozložiť na n neprázdnych podmnožín tak, že ak vyberieme ľubovoľný súčet $n-1$ čísel z $n-1$ ľubovoľných rôznych podmnožín, tak tento súčet patrí zostávajúcej n -tej podmnožine.

PIATA SÉRIA

- 5.1** Rovnica

$$x^4 + ax^3 + 2x^2 + bx + 1 = 0$$

s reálnymi koeficientami a, b má aspoň jeden reálny koreň. Nájdite najmenšiu možnú hodnotu súčtu $a^2 + b^2$.

- 5.2** Dané je prirodzené číslo n , $n > 2$, také, že n^2 možno zapísať ako rozdiel tretích mocnín dvoch za sebou idúcich prirodzených čísel. Dokážte, že n je súčtom dvoch štvorcov. Nájdite aspoň jedno také číslo n .

- 5.3** Nech r_1, r_2, \dots, r_m sú kladné racionálne čísla, pre ktoré platí $\sum_{k=1}^m r_k = 1$.

Definujme funkciu f predpisom

$$f(n) = n - \sum_{k=1}^m [r_k n] \quad \text{pre každé prirodzené číslo } n.$$

Určte najmenšiu a najväčšiu možnú hodnotu $f(n)$ ($\max_n f(n)$, $\min_n f(n)$) vzhľadom na r_1, \dots, r_m . (Symbolom $\lfloor x \rfloor$ označujeme dolnú celú časť čísla x .)

- 5.4** Trojuholník ABC je rovnoramenný ($|AB| = |AC|$). Označme D priesečník osi uhla ABC so stranou AC . Určte veľkosť uhla BAC , ak platí $|BC| = |BD| + |AD|$.
- 5.5** Daný je pravidelný štvorsten. Označme φ veľkosť najväčšieho zo šiestich uhlov, ktoré zvierajú jeho hrany s vodorovnou rovinou. Určte najmenšiu možnú hodnotu φ pre všetky možné polohy štvorstena vzhľadom na horizontálnu rovinu.
- 5.6** Dané je reálne číslo x , $x > 1$, ktoré nie je celé. Označme

$$a_n = \lfloor x^{n+1} \rfloor - x \lfloor x^n \rfloor, \quad n = 1, 2, \dots$$

Dokážte, že postupnosť čísel $\{a_n\}$ nie je periodická. (Symbolom $\lfloor x \rfloor$ označujeme dolnú celú časť čísla x .)

- 5.7** Na stranách AB a BC daného trojuholníku ABC sú dané po rade body X a N tak, že

$$|AX| \cdot |BN| = 2 \cdot |BX| \cdot |CN| \quad \text{a} \quad |\sphericalangle BNX| = |\sphericalangle ANC|.$$

Zistite, akú veľkosť môže mať uhol ACB .

Riešenia súťažných úloh KS SK MO

PRVÁ SÉRIA

1.1 (*Miroslav Dudík*) Najprv ukážeme, že z M musíme vylúčiť aspoň 5 prvkov. Budeme postupovať sporom. Predpokladajme teda, že stačí vylúčiť 4 prvky alebo menej.

Z trojíc 1, 4, 9; 2, 3, 6; 8, 7, 14; 12, 5, 15 musíme vylúčiť po jednom prvku (sú disjunktné a súčin čísel v každej z nich je druhá mocnina). Na základe predpokladu teda z každej z nich vylúčime práve jeden prvok a okrem nich už nemôžeme vylúčiť žiaden iný. Keďže nemožno vylúčiť číslo 10, treba vylúčiť jedno z čísel 6, 15 (pretože $10 \cdot 6 \cdot 15 = 30^2$).

Z 1, 4, 9 určite ostanú dva prvky. Ľubovoľný z nich vynásobený s $3 \cdot 12$, resp. $2 \cdot 8$ však dáva druhú mocninu. Teda musíme vylúčiť aspoň jeden z 2, 8 aj z 3, 12. V týchto dvojiciach sa nenachádza ani 7 ani 14. Musíme teda vylúčiť 8 (inak by bolo treba vylúčiť 7 alebo 14, čo by spolu s 1, resp. 4, resp. 9 dávalo aspoň 5 vylúčených). Teda 2 isto nemožno vylúčiť. V dvojiciach 2, 8 a 3, 12 sa nenachádzajú ani čísla 5, 10, t.j. aj tieto budú v hľadanej podmnožine. Avšak $2 \cdot 5 \cdot 10 = 10^2$, čo je spor s predpokladom.

Teraz ukážeme, že stačí vylúčiť 5 prvkov. Budú to napr. prvky 1, 2, 3, 8, 15. Ostala nám teda podmnožina $N = \{4, 5, 6, 7, 9, 10, 11, 12, 13, 14\}$. To, že táto podmnožina vyhovuje podmienkam úloh sa už ľahko overí.

Hľadaný počet prvkov je preto 10.

1.2 Platí

$$\begin{aligned} a^5 + b^5 &= (a + b)(a^4 - a^3b + a^2b^2 - ab^3 + b^4) = \\ &= (a + b)((a - b)^2(a^2 + ab + b^2) + a^2b^2) \geq \\ &\geq a^2b^2(a + b), \end{aligned}$$

rovnosť nastáva práve vtedy, keď $a = b$. Tento dôkaz sa dá urobiť aj nasledovne:

$$\begin{aligned} a^5 + b^5 - a^3b^2 - a^2b^3 &\geq 0, \\ a^3(a^2 - b^2) + b^3(b^2 - a^2) &\geq 0, \\ (a^3 - b^3)(a^2 - b^2) &\geq 0. \end{aligned}$$

všetky úpravy boli ekvivalentné a posledná nerovnosť je pre dvojicu kladných reálnych čísel triviálna.

Preto

$$\begin{aligned} \frac{ab}{a^5 + b^5 + ab} &\leq \frac{ab}{a^2b^2(a + b) + ab} = \frac{1}{ab(a + b) + 1} = \\ &= \frac{1}{ab(a + b + c)} = \frac{c}{(a + b + c)}. \end{aligned}$$

Potom vzhľadom na cyklickú zámenu

$$\frac{ab}{a^5 + b^5 + ab} + \frac{bc}{b^5 + c^5 + bc} + \frac{ca}{c^5 + a^5 + ca} \leq \frac{c}{(a+b+c)} + \frac{a}{(a+b+c)} + \frac{b}{(a+b+c)} = 1,$$

čo bolo treba dokázať. Rovnosť zrejme nastáva práve vtedy, keď $a = b = c = 1$.

1.3 Najprv popíšeme stratégiu prvého hráča (ďalej \mathcal{H}_1) zaručujúcu $A \geq 6$ nezávisle na hre súpera.

1.ťah: „1“ \rightarrow C3 (obr. 1). Bez ujmy na všeobecnosti druhý hráč vpíše „0“ do jedného zo štvorčekov A1, A2, A3, B2 alebo B3 (pozri vyšrafovaný útvar na obr. 1).

2.ťah: „1“ \rightarrow D3. Hráč \mathcal{H}_2 je teraz nútený vpísať „0“ do štvorca 3×3 s ľavým horným rohom v C5 (ďalej budeme takýto štvorec označovať S_{C5}). Inak by totiž \mathcal{H}_1 ľahko na zvyšných sedem políčkoch v tomto štvorci vpísal aspoň štyri „1“, čím by v ňom získal aspoň šesť „1“. Analogická je situácia aj v štvorci S_{C3} (obr. 1). Keďže jediné voľné políčko v prieniku týchto štvorcov je E3, potom druhý hráč musí ťahať „0“ \rightarrow E3.

3.ťah: „1“ \rightarrow C4. V každom zo štvorcov S_{B5} , S_{B4} , S_{C5} , S_{C4} sú teraz tri „1“ a najviac jedna „0“ (obr. 2). Pokiaľ by \mathcal{H}_2 nevpísal „0“ do hociktorého z týchto štvorcov, \mathcal{H}_1 by na zvyšných aspoň päť voľných miest poľahky vpísal aspoň tri „1“, čím by v tomto štvorci bolo aspoň šesť „1“. Preto \mathcal{H}_2 musí napísať „0“ do políčka, ktoré je prienikom všetkých štyroch spomínaných štvorcov. Preto \mathcal{H}_2 musí ťahať „0“ \rightarrow D4.

4.ťah: „1“ \rightarrow E4. Teraz už \mathcal{H}_2 očividne nezabráni $A \geq 6$. \mathcal{H}_1 totiž určite nezávisle na hre \mathcal{H}_2 obsadí aspoň tri z políčkoch C5, D5, E5, C2, D2 a E2, čo znamená, že aspoň do jedného zo štvorcov S_{C5} alebo S_{C4} dopíše aspoň dve „1“ k terajším štyrom (obr. 3).

5					
4					
3			1	1	
2					
1					
	A	B	C	D	E

Obr. 1

5					
4			1		
3			1	1	0
2					
1					
	A	B	C	D	E

Obr. 2

5			*	*	*
4			1	0	1
3			1	1	0
2			*	*	*
1					
	A	B	C	D	E

Obr. 3

5					
4					
3					
2					
1					
	A	B	C	D	E

Obr. 4

Preto $A \geq 6$. Nemožno však zaručiť aj $A \geq 7$? Nemožno, pretože \mathcal{H}_2 môže hrať tak, aby A bolo najviac 6.

Rozdelíme si šachovnicu 5×5 na 12 disjunktných obdĺžnikov 2×1 a jeden štvorček 1×1 (obr. 4). Ak \mathcal{H}_1 vpíše do ľubovoľného prázdneho obdĺžnika „1“, \mathcal{H}_2 vpíše do druhého štvorčeka toho istého obdĺžnika „0“. Pokiaľ \mathcal{H}_1 vpíše „1“ do stredného štvorčeka alebo do obdĺžnika, kde je už jedna „0“, vpíše \mathcal{H}_2 „0“ do ľubovoľného prázdneho políčka. Po ukončení hry bude takto určite v každom obdĺžniku práve jedna „0“. Keďže však každý z deviatich štvorcov 3×3 obsahuje aspoň tri celé obdĺžniky, sú v každom aspoň tri „0“, teda maximálne šesť „1“, teda $A \leq 6$.

Týmto sme dokázali $A = 6$.

1.4 Prepišeme si funkcionálnu rovnicu do iného tvaru:

$$f\left(x + \frac{1}{6} + \frac{1}{7}\right) - f\left(x + \frac{1}{7}\right) = f\left(x + \frac{1}{6}\right) - f(x).$$

Ak si definujeme novú funkciu $g(x) = f\left(x + \frac{1}{6}\right) - f(x)$, dostávame rovnosť:

$$g\left(x + \frac{1}{7}\right) = g(x).$$

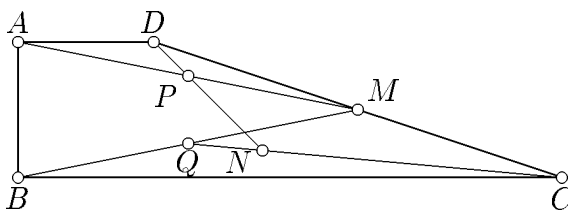
Teda $g(x)$ je periodická s periódou $\frac{1}{7}$ a nech n je ľubovoľné celé číslo. Potom zrejme $g(y + n) = g(y)$ pre všetky $x \in \mathbb{R}$. Teraz už iba upravujeme:

$$\begin{aligned} f(x + n + 1) - f(x + n) &= \\ &= \left(f(x + n + 1) - f\left(x + n + \frac{5}{6}\right)\right) + \left(f\left(x + n + \frac{5}{6}\right) - f\left(x + n + \frac{4}{6}\right)\right) + \\ &\quad + \left(f\left(x + n + \frac{4}{6}\right) - f\left(x + n + \frac{3}{6}\right)\right) + \left(f\left(x + n + \frac{3}{6}\right) - f\left(x + n + \frac{2}{6}\right)\right) + \\ &\quad + \left(f\left(x + n + \frac{2}{6}\right) - f\left(x + n + \frac{1}{6}\right)\right) + \left(f\left(x + n + \frac{1}{6}\right) - f(x + n)\right) = \\ &= g\left(x + n + \frac{5}{6}\right) + g\left(x + n + \frac{4}{6}\right) + g\left(x + n + \frac{3}{6}\right) + \\ &\quad + g\left(x + n + \frac{2}{6}\right) + g\left(x + n + \frac{1}{6}\right) + g(x + n) = \\ &= g\left(x + \frac{5}{6}\right) + g\left(x + \frac{4}{6}\right) + g\left(x + \frac{3}{6}\right) + g\left(x + \frac{2}{6}\right) + g\left(x + \frac{1}{6}\right) + g(x) = h(x) \end{aligned}$$

Dokázali sme teda, že $f(x + n + 1) - f(x + n) = h(x)$, a to pre všetky $n \in \mathbb{Z}$. Ak by $h(x)$ bolo rôzne od 0, došli by sme k sporu – funkčné hodnoty $f(x)$ by už neboli v intervale $\langle -1, 1 \rangle$ (platí totiž $f(x + n) = f(x) + n \cdot h(x)$, čo pre nenulové $h(x)$ rastie do nekonečna). Preto nevyhnutne musí platiť $f(x + n + 1) - f(x + n) = 0$, čo nie je nič iné ako to, že funkcia $f(x)$ je periodická s periódou 1.

1.5 Uvedené tvrdenie neplatí. Zrejme stačí nájsť aspoň jeden kontrapríklad. Zvoľme súradnice bodov nasledovne: $A = [0; 1]$, $B = [0; 0]$, $C = [4; 0]$, $D = [1; 1]$, ďalej $M = [2, 5; 0, 5]$, $P = [1, 25; 0, 75]$, $Q = [1, 25; 0, 25]$. Súradnice bodu $N = [1, 8; 0, 2]$ ľahko

spočítame ako prienik dvoch priamok (obr. 5). Bod N je pod priamkou BM , čiže mimo trojuholníka ABM .



Obr. 5

1.6 (Vladimír Marko) Ak p a q nie sú nesúdeliteľné, môžeme p, q aj $x_i, i = 0, 1, \dots, n$ vydeliť najväčším spoločným deliteľom čísel p a q a prevedieme celý problém na riešenie s p, q nesúdeliteľnými.

Teraz už môžeme predpokladať, že p, q sú nesúdeliteľné. Ak posledný člen postupnosti x_n vznikol k -násobným pripočítaním čísla p a l -násobným odpočítaním čísla q , potom $0 = x_n = kp - lq$, kde $k + l = n$. Teda triviálne $q|k$ a $p|l$. Nech $k = k_0q$ a $l = l_0p$. Potom $x_n = (k_0 - l_0)pq$, čiže $k_0 = l_0$. Z toho $n = k + l = k_0(p + q)$.

Ďalej označme $f_i = x_i - x_{i-p-q}$ pre $i = p + q, p + q + 1, \dots, n$. Ak niektoré z týchto $f_i = 0$, potom $x_i = x_{i-p-q}$ a tvrdenie zrejme platí. Preto môžeme predpokladať, že všetky $f_i \neq 0$. Člen f_{p+q} možno napísať v tvare $f_{p+q} = (q - a_{p+q})p - (p + a_{p+q})q$, pre vhodné celé číslo a_{p+q} (pretože $q - a_n + p + a_n = q + p$). Toto po úprave dáva $f_{p+q} = -a_{p+q}(p + q)$. Keďže $f_{p+q} \neq 0$, potom aj $a_{p+q} \neq 0$. Ďalej

$$f_i = f_{i-1} + x_i - x_{i-1} - x_{i-p-q} + x_{i-p-q-1}, \quad \text{pre } i = p + q + 1, \dots, n.$$

Môžu preto nastať len tri možnosti

$$f_i = f_{i-1} - (p + q), \quad f_i = f_{i-1}, \quad f_i = f_{i-1} + (p + q).$$

Keďže však vieme, že f_{p+q} je tvaru $-a_{p+q}(p+q)$, potom aj každé $f_i = -a_i(p+q)$, pre $i = p+q, \dots, n$. Predchádzajúce tri možnosti sa menia na $a_i = a_{i-1} + c$, kde $c \in \{-1, 0, 1\}$. Zrejme sú všetky $a_i \neq 0$. Potom však musia mať stále rovnaké znamienko. Rovnaké znamienko majú potom aj všetky f_i , čo ale znamená, že $x_n = f_{p+q} + f_{2(p+q)} + \dots + f_{k_0(p+q)}$, kde každý člen súčtu má rovnaké znamienko. To je však spor s predpokladom $x_n = 0$. Preto musí existovať nejaké vhodné i také, že $x_i = x_{i-p-q}$. Keďže $n - p - q \neq 0$, je táto dvojica indexov rôzna od $(0, n)$.

Iné riešenie. (Viera Růžičková)

Pre jednoduchosť opäť uvažme prípad p, q nesúdeliteľné. Uvažujme nasledujúcu tabuľku:

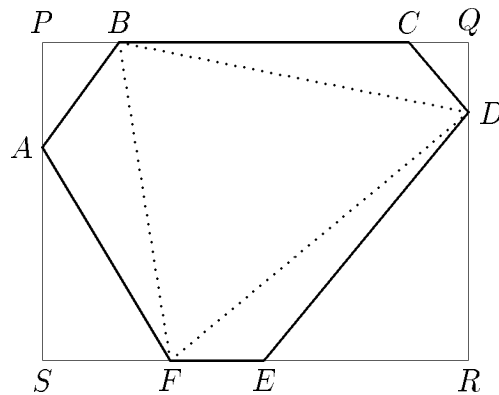
- do ľavého dolného rohu umiestnime číslo 0;
- do políčka vzdialeného o k polí vpravo a o l polí hore od ľavého dolného rohu umiestnime číslo $kp - lq$.

Postupnosť x_0, x_1, \dots, x_n si možno predstaviť v tejto tabuľke ako lomenú čiaru vychádzajúcu z ľavého dolného rohu vedúcu do nejakého políčka, v ktorom je nula (okrem políčka v $(q+1)$ -vom stĺpci a $(p+1)$ -vom riadku – pretože $n > p+q$). Všetky nuly v tabuľke zrejme ležia na jednej priamke, prechádzajúcej ľavým dolným rohom. Ďalej si môžeme všimnúť, že v tabuľke sa vždy periodicky opakuje podtabuľka, ktorej ľavý dolný roh a pravý horný roh ohraničujú dve za sebou idúce nuly. Prvá takáto podtabuľka má ľavý dolný roh v ľavom dolnom rohu celej tabuľky a pravý horný roh v políčku v $(q+1)$ -vom stĺpci a $(p+1)$ -vom riadku (v prípade p a q nesúdeliteľných).

Teraz postupujme nasledovne. V tabuľke zaznačujeme členy postupnosti x_i pohybom vždy buď vpravo alebo nahor. Pri prechode na nové políčko vždy v tabuľke preškrtneme všetky políčka, na ktorých sú čísla rovnaké, ako je číslo na novom políčku. Na dôkaz tvrdenia v zadaní potrebujeme dokázať, že vždy, keď chceme lomenú čiaru ukončiť v políčku s nulou, musíme aspoň raz vstúpiť na preškrtnuté políčko. Keďže lomená čiara nemôže skončiť v prvej podtabuľke, musí ju raz nadobro opustiť. Bez ujmy na všeobecnosti nech ju opustí pod ďalšou podtabuľkou (mohla aj nad ňou). To však znamená, že v každom stĺpci podtabuľky je už vyškrtnuté aspoň jedno políčko. Potom však nemôže existovať lomená čiara neprechádzajúca preškrtnutým políčkom vedúca do políčka s nulou, pretože nula sa nachádza vždy až v najvyššom riadku každej podtabuľky a je (ak už nebola vyškrtnutá nula v $(q+1)$ -vom stĺpci a $(p+1)$ -vom riadku – potom je hľadaná dvojica indexov $(0, p+q)$) „odrezaná“ od aktuálnej polohy neprerušenu lomenou čiarou vedúcou pod ňou. Preto sa v postupnosti nutne musí nachádzať dvojica členov s rovnakou hodnotou rôzna od dvojice (x_0, x_n) .

1.7 Označme po rade a, b, c, d, e a f dĺžky strán AB, BC, CD, DE, EF a FA . Keďže zo zadania vyplýva, že protilahlé uhly šesťuholníka sú zhodné, označme ich nasledovne: $\varphi = |\sphericalangle FAB| = |\sphericalangle CDE|$, $\psi = |\sphericalangle ABC| = |\sphericalangle DEF|$ a $\omega = |\sphericalangle BCD| = |\sphericalangle EFA|$ (obr. 6). Uvažujme body P, Q, R, S tak, aby $AP \perp BC$, $AS \perp EF$, $DQ \perp BC$, $DR \perp EF$ a zároveň P, Q ležali na priamke BC a R, S na priamke EF . Potom $PQRS$ je pravouholník a $|BF| \geq |PS| = |QR|$. Preto $2|BF| \geq |PS| + |QR|$, čiže

$$2|BF| \geq (a \sin \psi + f \sin \omega) + (c \sin \omega + d \sin \psi). \quad (1)$$



Obr. 6

Cyklickou zámennou dostávame ďalej

$$2|DB| \geq (c \sin \varphi + b \sin \psi) + (e \sin \psi + f \sin \varphi), \quad (2)$$

$$2|FD| \geq (e \sin \omega + d \sin \varphi) + (a \sin \varphi + b \sin \omega). \quad (3)$$

Pre polomery kružníc opísaných trojuholníkom FAB , BCD a DEF platí

$$R_A = \frac{|BF|}{2 \sin \varphi}, \quad R_C = \frac{|DB|}{2 \sin \omega}, \quad R_E = \frac{|FD|}{2 \sin \psi}. \quad (4)$$

Odtiaľ po dosadení za R_A, R_B a R_C z (3) a potom za $|BF|, |DB|$ a $|FD|$ z (1), (2) a (3):

$$\begin{aligned} & 4(R_A + R_C + R_E) \geq \\ & \geq a \left(\frac{\sin \psi}{\sin \varphi} + \frac{\sin \varphi}{\sin \psi} \right) + b \left(\frac{\sin \psi}{\sin \omega} + \frac{\sin \omega}{\sin \psi} \right) + c \left(\frac{\sin \omega}{\sin \varphi} + \frac{\sin \varphi}{\sin \omega} \right) + \\ & + d \left(\frac{\sin \varphi}{\sin \psi} + \frac{\sin \psi}{\sin \varphi} \right) + e \left(\frac{\sin \psi}{\sin \omega} + \frac{\sin \omega}{\sin \psi} \right) + f \left(\frac{\sin \varphi}{\sin \omega} + \frac{\sin \omega}{\sin \varphi} \right). \end{aligned}$$

Po využití známej nerovnosti

$$\frac{x}{y} + \frac{y}{x} \geq 2,$$

pre kladné čísla x a y dostávame

$$4(R_A + R_C + R_E) \geq 2(a + b + c + d + e + f) = 2P,$$

a teda $R_A + R_C + R_E \geq \frac{P}{2}$, čo bolo treba dokázať.

Poznámka. Rovnosť platí, ak $\varphi = \psi = \omega$ a zároveň $BF \perp BC, \dots$; teda práve vtedy, keď je šesťuholník pravidelný.

DRUHÁ SÉRIA

2.1 (Ján Špakula)

a) Dokážeme sporom, že neexistuje $n \in \mathbb{N}$ také, že $S(n) = n^2 - 13$.

Nech také n existuje. Všimnime si zápis n^2 ako súčet $n^2 - 13$ štvorcov. Nech $n^2 - 13 - k$ z nich je jednotiek. Potom pomocou k štvorcov (rôznych od 1) musíme zapísať číslo $n^2 - (n^2 - 13 - k) = k + 13$. Poľahky sa overí, že pre $k = 1, 2, 3, 4$ sa to nedá. Pre $k \geq 5$ však platí $4k > k + 13$, ale súčet k štvorcov rôznych od 1 (každý štvorec je aspoň 4) je najmenej $4k$, čo je spor.

b) Riešením je $n = 13$, čiže $S(13) = 13^2 - 14 = 155$. Všimnime si najprv množinu \mathcal{H} čísel (štvorcov) $\{1, 4, 16, 25, 49, 64, 100, 121, 169\}$. Pre čísla z tejto množiny platia nasledujúce vzťahy:

$$\begin{aligned} 4 &= 1 + 1 + 1 + 1; & 64 &= 16 + 16 + 16 + 16; \\ 16 &= 4 + 4 + 4 + 4; & 100 &= 25 + 25 + 25 + 25; \\ 25 &= 16 + 4 + 4 + 1; & 121 &= 100 + 16 + 4 + 1; \\ 49 &= 16 + 16 + 16 + 1; & 169 &= 100 + 49 + 16 + 4. \end{aligned}$$

Z týchto rovností vyplýva, že každé číslo z množiny $\mathcal{H} \setminus \{1\}$ vieme napísať ako súčet štyroch čísel z \mathcal{H} . Preto vieme aj číslo 169 zapísať ako súčet $4, 7, 11, \dots, 3k+1, \dots, 169$ štvorcov z \mathcal{H} (kým nemáme samé jednotky môžeme ľubovoľný sčítanec rozpísať ako súčet štyroch ďalších).

Keďže navyše platia aj rovnosti

$$169 = 144 + 16 + 9, \quad 144 = 100 + 36 + 4 + 4, \quad 36 = 25 + 9 + 1 + 1,$$

vieme 169 napísať aj ako súčet $3, 6, 9, \dots, 3k+3, \dots, 153$ štvorcov ($169 = 144 + 16 + 9 = 100 + 36 + 16 + 9 + 4 + 4 = (9 + 9) + 100 + 25 + 16 + 4 + 4 + 1 + 1 = \dots$, kde čísla 100, 25, 16 a 4 sú už z množiny \mathcal{H}).

Napokon z rovností

$$169 = 144 + 25, \quad 144 = 100 + 36 + 4 + 4, \quad 36 = 25 + 9 + 1 + 1$$

a predchádzajúcich dostávame, že 169 sa dá napísať aj ako súčet $2, 5, \dots, 3k+2, \dots, 161$ štvorcov ($169 = 144 + 25 = 100 + 36 + 25 + 4 + 4 = 9 + 100 + 25 + 25 + 4 + 4 + 1 + 1 = \dots$, kde čísla 100, 25 a 4 sú už z množiny \mathcal{H}).

Z toho už vyplýva, že 169 sa dá napísať ako súčet $1, 2, \dots, 155$ štvorcov, a z časti a) vyplýva, že sa nedá napísať ako súčet 156 štvorcov, a preto $S(13) = 155$.

c) Ukážeme, že riešením časti b) sú aj všetky čísla n tvaru $n = 2^k \cdot 13$, $k = 0, 1, \dots$. Tým dokážeme tvrdenie c).

Množinu \mathcal{H}' voľme nasledujúco

$$\mathcal{H}' = \{1, 2 \cdot 2^m, 5 \cdot 2^m, 7 \cdot 2^m, 11 \cdot 2^m, 13 \cdot 2^m; m = 0, 1, 2, \dots\}.$$

Platia rovnosti:

$$\begin{aligned} (2 \cdot 2^m)^2 &= 4^m + 4^m + 4^m + 4^m, \\ (5 \cdot 2^m)^2 &= 16 \cdot 4^m + 4 \cdot 4^m + 4 \cdot 4^m + 4^m, \\ (7 \cdot 2^m)^2 &= 16 \cdot 4^m + 16 \cdot 4^m + 16 \cdot 4^m + 4^m, \\ (11 \cdot 2^m)^2 &= 100 \cdot 4^m + 16 \cdot 4^m + 4 \cdot 4^m + 4^m, \\ (13 \cdot 2^m)^2 &= 100 \cdot 4^m + 49 \cdot 4^m + 16 \cdot 4^m + 4 \cdot 4^m. \end{aligned}$$

Opäť každé číslo z $\mathcal{H}' \setminus \{1\}$ vieme napísať ako súčet štyroch čísel z \mathcal{H}' . Použitím rozkladov

$$\begin{aligned}(13 \cdot 2^m)^2 &= 9 \cdot 4 \cdot 4^m + 4 \cdot 4^m + 9 \cdot 4^m, \\ (9 \cdot 2^m)^2 &= 36 \cdot 2^{2(m-1)} = 25 \cdot 4^{m-1} + 9 \cdot 4^{m-1} + 4^{m-1} + 4^{m-1}, \\ (13 \cdot 2^m)^2 &= 9 \cdot 4 \cdot 4^m + 25 \cdot 4^m\end{aligned}$$

pre $m = 0, 1, \dots$ dostávame úplne rovnako ako v časti b), že číslo $n = 13 \cdot 2^k$ môžeme zapísať postupne

- ako súčet $1, 4, 7, \dots, n^2$ štvorcov,
- ako súčet $3, 6, 9, \dots, n^2 - 16$ štvorcov (na konci bude $9 + 9 + (n^2 - 18) \times 1$),
- ako súčet $2, 5, 8, \dots, n^2 - 8$ štvorcov (na konci bude $9 + (n^2 - 9) \times 1$).

Teda $n = 13 \cdot 2^k$ sa dá napísať ako súčet $1, 2, \dots, n^2 - 14$ štvorcov. Podľa a) nemožno použiť práve $n^2 - 13$ štvorcov, čiže $S(13 \cdot 2^k) = (13 \cdot 2^k)^2 - 14$.

2.2 Označme s_i a v_i po rade šírku a výšku i -teho obdĺžnika ($i = 1, 2, \dots, 100$) tak, aby $s_i \geq v_i$ pre každé i . Roztriedme teraz obdĺžniky nasledovne: obdĺžnik dáme do triedy k práve vtedy, keď $s_i - v_i = 2k$ alebo $s_i - v_i = 2k + 1$, $k = 0, 1, \dots, 49$. Spolu teda máme práve 50 tried. (Ak si predstavíme tieto obdĺžniky ako mrežové body v rovine a stotožníme obdĺžnik i s mrežovým bodom (s_i, v_i) , tak máme 101 mrežových bodov v trojuholníku s vrcholmi $(0, 0)$, $(100, 0)$ a $(100, 100)$, body môžu ležať aj na prepone. Tento trojuholník sme triedami rozdelili na 50 pruhov priamkami spájajúcimi body $(2k, 0)$ a $(100, 100 - 2k)$, $k = 1, 2, \dots, 49$.) Keďže máme 101 obdĺžnikov v 50 triedach, jedna z tried obsahuje aspoň 3 obdĺžniky. Nech sú označené \mathcal{X} , \mathcal{Y} a \mathcal{Z} po rade podľa rastúcej šírky ($s_{\mathcal{X}} \leq s_{\mathcal{Y}} \leq s_{\mathcal{Z}}$). Ak sa nejaké dve šírky zhodujú, označíme ich podľa rastúcej výšky.

Najprv dokážeme, že \mathcal{X} možno umiestniť do \mathcal{Y} . Ak by totiž náhodou $v_{\mathcal{X}} > v_{\mathcal{Y}}$, potom musí byť $s_{\mathcal{X}} = s_{\mathcal{Y}}$, aby mohli \mathcal{X} a \mathcal{Y} byť v tej istej triede (ak $s_{\mathcal{Y}} > s_{\mathcal{X}}$, potom sa čísla $s_{\mathcal{Y}} - v_{\mathcal{Y}}$ a $s_{\mathcal{X}} - v_{\mathcal{X}}$ líšia najmenej o dve, pretože všetky dĺžky aj šírky sú celočíselné), čo je spor so systémom nášho označenia. Rovnakým spôsobom sa ukáže, že \mathcal{Y} možno umiestniť požadovaným spôsobom do \mathcal{Z} . Tým je zadané tvrdenie dokázané.

2.3 Nech body P, Q a R vyhovujú zadaniu. Zavedme označenie ako na obr. 7. Najprv ukážeme, že trojuholníky SMP a SMV majú rovnaký obsah. Platí:

$$S_{SAP} + S_{SAQ} + S_{SVR} = \frac{1}{2} S_{PQR} = S_{PQM}. \quad (1)$$

Vyplýva to z toho, že body A, B a V sú dotykové body kružnice k s jednotlivými stranami trojuholníka PQR a z toho, že PM je ťažnica tohto trojuholníka.

Bod M je stred QR , a preto platí :

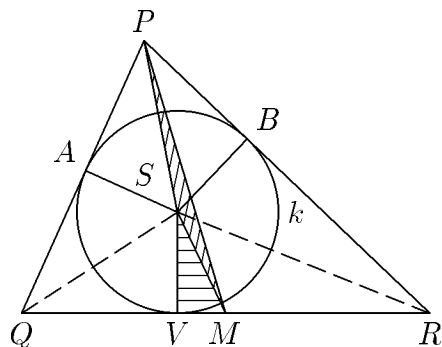
$$S_{SQV} + S_{SMV} = S_{SMR}.$$

Teda aj

$$S_{SQV} + 2S_{SMV} = S_{SMR} + S_{SMV} = S_{SVR}.$$

Dosadením do (1) dostaneme rovnosť:

$$S_{SAP} + S_{SAQ} + S_{SQV} + 2S_{SMV} = S_{PQM} = S_{SAP} + S_{SAQ} + S_{SQV} + S_{SMV} + S_{SMP}.$$



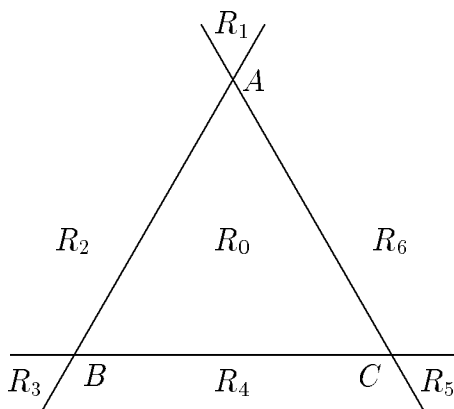
Obr. 7

Dokázali sme, že obsahy trojuholníkov SMV a SMP sú rovnaké. Nakoľko majú tieto trojuholníky spoločnú základňu SM , sú vzdialenosti bodov P a V od priamky SM rovnaké. Keďže každá výška trojuholníka musí byť väčšia ako dvojnásobok polomeru vpísanej kružnice a bod P musí ležať v polrovine QRS , bod P leží na polpriamke rovnobežnej s polpriamkou MS (rovnakej orientácie) začínajúcej v bode W . Bod W leží na kružnici k tak, že úsečka VW je jej priemer.

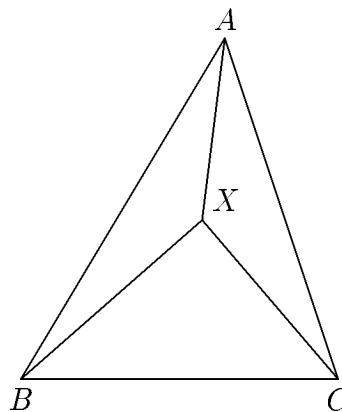
Poľahky sa presvedčíme, že všetky body tejto polpriamky vyhovujú zadaniu – spätným postupom skonštruujeme body Q, R, A a B ; ukážeme, že $S_{SMP} = S_{SMV}$, a že M je stred QR .

2.4 Ak body A, B a C ležia na jednej priamke, potom $m(ABC) = 0$, teda pre každý bod X roviny daná nerovnosť triviálne platí.

Preto môžeme predpokladať, že body A, B a C neležia na jednej priamke. Rozdelíme rovinu na 7 častí (oblastí) R_0, \dots, R_6 ako na obr.8. Rozoberieme niekoľko prípadov v závislosti na polohe bodu X .



Obr. 8



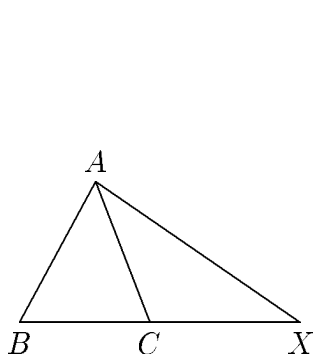
Obr. 9

Prípád 1. Nech bod X leží v oblasti R_0 , čiže vo vnútri trojuholníka ABC (obr. 9). Môžeme bez ujmy na všeobecnosti predpokladať, že $|BC| \geq |AC|$ a $|BC| \geq |AB|$. Preto je najkratšou výškou v trojuholníku ABC výška na stranu BC . Všimnime si, že všetky vzdialenosti $|AX|$, $|BX|$ a $|CX|$ sú kratšie ako $|BC|$. Ďalej pre obsahy trojuholníkov platí $S_{ABC} = S_{BXC} + S_{CXA} + S_{AXB}$. Predelením tejto rovnosti výrazom $\frac{|BC|}{2}$ a použitím toho, že BC je najdlhšou stranou trojuholníkov CXA a AXB , dostávame nerovnosť (1).

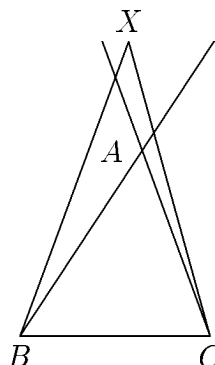
Prípád 2. Nech bod X leží na jednej zo strán trojuholníka ABC . Ak X patrí obvodu trojuholníka ABC , môžeme nerovnosť (1) dokázať tak isto ako v prvom prípade. Preto predpokladajme, že X leží na predĺžení jednej zo strán, napr. BC (obr. 10). Potom $m(ABC) \leq m(AXB)$. (To možno dokázať prešetrovaním viacerých prípadov pre $\triangle ABC$: ak je najkratšia výška trojuholníka ABC výška na stranu AC alebo AB , tvrdenie je zrejmé; ak je to výška na stranu BC , potom pre uhly platí $|\sphericalangle CAB| > |\sphericalangle ABC|$ a $|\sphericalangle CAB| > |\sphericalangle BCA|$, z čoho vyplýva, že vzdialenosť bodu B od priamky AX je väčšia ako vzdialenosť bodu A od priamky BC .) Platí tiež $m(BXC) = 0$. Teda

$$m(ABC) \leq m(AXB) \leq m(BXC) + m(CXA) + m(AXB).$$

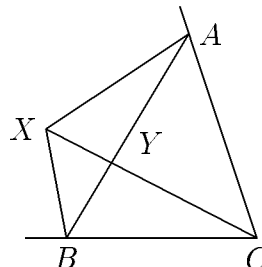
Prípád 3. Nech X leží v jednej z oblastí R_1, R_3 alebo R_5 , napr. R_1 (obr. 11). Potom A leží vnútri trojuholníka BXC , čiže opäť $m(BXC) \geq m(ABC)$. Z toho okamžite vyplýva (1).



Obr. 10



Obr. 11



Obr. 12

Prípád 4. Napokon nech X leží v ľubovoľnej zostávajúcej oblasti, napr. R_2 (obr. 12). Označme Y priesečník úsečiek AB a CX . Potom pre trojuholník CYA a bod X nastáva prípad 2, čiže $m(CYA) \leq m(AXC)$ a $m(BYC) \leq m(BXC)$. Z toho

$$m(ABC) \leq m(AYB) + m(BYC) + m(CYA) \leq m(AXB) + m(BXC) + m(CXA),$$

pretože $m(AYB) = 0 \leq m(AXB)$.

Prešetrili sme všetky možné prípady polohy bodu X , tvrdenie úlohy je dokázané.

2.5 (*Pavol Novotný*) Najskôr dokážeme sporom, že pre čísla n deliteľné tromi sa hra nemôže skončiť s jedným mydlom na šachovnici. Očíslujme políčka tromi číslami: 1, 2 a 3 tak, aby vždy nad každou jednotkou bola dvojka, nad každou dvojkou trojka, nad každou trojkou jednotka a vpravo od jednotky trojka, vpravo od trojky dvojka

a napokon vpravo od dvojky jednotka (prvú jednotku môžeme zatiaľ zvoliť ľubovoľne). Potom vo štvorci s mydlami so stranou $n = 3k$, $k = 1, 2, \dots$ je rovnaký počet polí očíslovaných každým z čísel 1, 2 a 3. Súčet všetkých čísel polí s mydlami je teda $(1 + 2 + 3) \cdot \frac{n^2}{3} = 2n^2$, čiže párný. Jednoducho sa ukáže, že každý možný ťah zachováva paritu súčtu čísel, na ktorých sú umiestnené mydlá (stačí preveriť 6 situácií). Ak má teda na konci zostať práve jedno mydlo, musí byť na poli s číslom 2. Šachovnicu však môžeme očíslovať rovnakým spôsobom, len prvú jednotku zvolíme na políčku, ktoré má teraz číslo 2. Potom použitím rovnakých úvah možno dokázať, že políčko, na ktorom zostane posledné mydlo musí mať aj v tomto očíslovaní číslo 2, to však nie je možné.

Teraz dokážeme (nájdeme postup), že pre každé prirodzené číslo n nedeliteľné tromi sa hra môže skončiť s jedným mydlom. Najprv zistíme, že hra sa tak môže skončiť pre $n = 1$ a $n = 2$ (ľahko sa nájde postupnosť ťahov). Nasledujúcou postupnosťou ťahov dokážeme zo šachovnice odstrániť tri mydlá, ktoré ležia vedľa seba (v jednom riadku alebo stĺpci) tak, aby ostatné mydlá zostali na svojich miestach (obr. 13):



Obr. 13

Teda, na takéto odstránenie troch mydiel vedľa seba sme potrebovali len to, aby pri prvom (resp. treťom) mydle ležalo jedno mydlo a na jeho opačnej strane bolo voľné políčko. Stačí už len ukázať, že každý štvorec $n \times n$ vieme dostať zo štvorca 2×2 a trojíc susedných políčok spĺňajúcich predchádzajúce podmienky. Podľa obr. 14 vieme takto vytvoriť štvorce 4×4 a 5×5 .



Obr. 14

Obdobným spôsobom, ako sme dostali zo štvorca 2×2 štvorec $n \times n$, zrejme dostaneme aj z každého štvorca $m \times m$ štvorec $(m+3) \times (m+3)$. Potom vyššie popísanou postupnosťou ťahov odstránime najprv trojice susedných kameňov (smerom dovnútra), a napokon štvorec 2×2 .

2.6 (Peter Bodík, Peter Kozák, Pavol a Peter Novotný a Ján Špakula)

Označme S priesečník osí CE a BD (obr. 15) a uhly trojuholníka ABC ako zvyčajne. Potom zrejme $|\sphericalangle BSC| = 138^\circ$ a z trojuholníka BSC dostávame $\frac{\beta}{2} + \frac{\gamma}{2} = 84^\circ$, čiže $\alpha =$

$= 96^\circ$. Použitím sínusových viet pre trojuholníky ESD , DSA a ASE dostávame

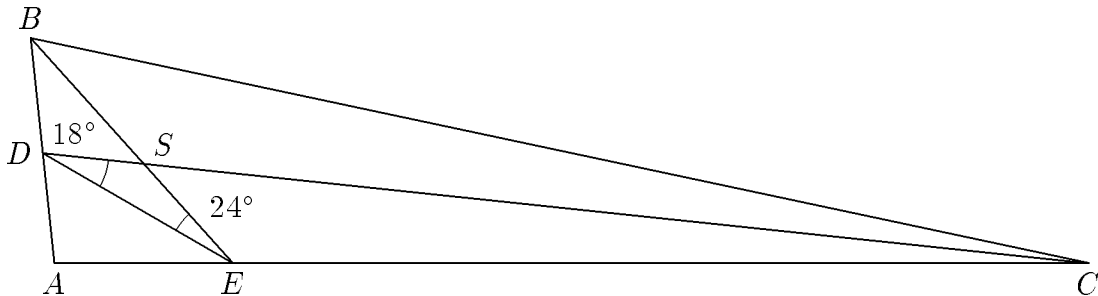
$$|DS| = \frac{|ES| \cdot \sin 28^\circ}{\sin 24^\circ}, \quad |AS| = \frac{|DS| \cdot \sin \left(84^\circ - \frac{\beta}{2}\right)}{\sin 48^\circ}, \quad |AS| = \frac{|ES| \cdot \sin \left(42^\circ + \frac{\beta}{2}\right)}{\sin 48^\circ}.$$

Spojením týchto vzťahov a vylúčením dĺžok úsečiek dostávame goniometrickú rovnicu

$$\sin 18^\circ \cdot \sin \left(84^\circ - \frac{\beta}{2}\right) = \sin 24^\circ \cdot \sin \left(42^\circ + \frac{\beta}{2}\right).$$

Teraz stačí rozpísať výrazy $\sin \left(84^\circ - \frac{\beta}{2}\right)$ a $\sin \left(42^\circ + \frac{\beta}{2}\right)$ podľa súčtových vzorcov a po úprave dostaneme rovnicu:

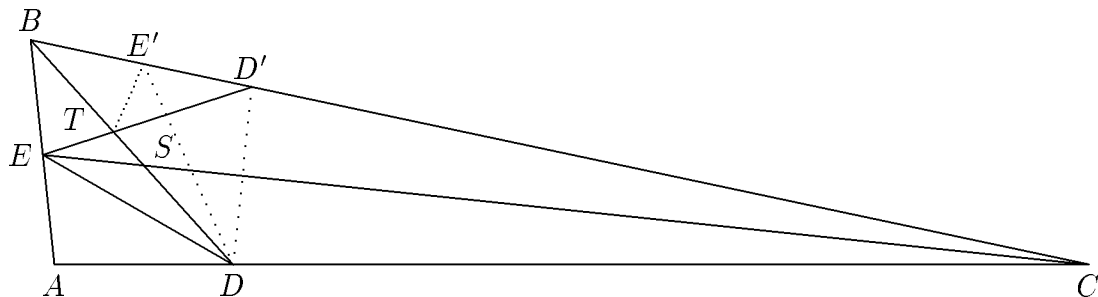
$$\cos \frac{\beta}{2} \cdot (\sin 84^\circ \cdot \sin 18^\circ - \sin 42^\circ \cdot \sin 24^\circ) = \sin \frac{\beta}{2} \cdot (\cos 84^\circ \cdot \sin 18^\circ + \cos 42^\circ \cdot \sin 24^\circ). \quad (1)$$



Obr. 15

Vyjadrením tangensu uhla $\frac{\beta}{2}$ a spočítaním (napr. pomocou kalkulačky) hodnoty výrazu dostávame $\beta = 12^\circ$, a potom $\gamma = 72^\circ$. Pravda, na presné určenie by bolo potrebné goniometrický výraz buď zjednodušiť pomocou súčtových vzťahov, alebo vyčíslit hodnoty $\sin 12^\circ$ a $\cos 12^\circ$, pomocou ktorých už vieme spočítať všetky číselné hodnoty v (1) a overiť, že $\beta = 12^\circ$ je riešením. Potom z monotónnosti oboch strán v (1) (vzhľadom na β) zistíme, že je to naozaj jediné riešenie.

Iné riešenie. Najprv určíme uhol α ako v prvom riešení. Označme E' a D' po rade obrazy bodov E a D v osových súmernostiach po rade podľa BD a CE (zrejme obidva body ležia na BC). Ďalej označme T priesečník BD a ED' a spojme TE' , DD' a DE' (obr. 16).



Obr. 16

Zo symetrie $|\sphericalangle CED'| = |\sphericalangle CED| = 18^\circ$ a $|\sphericalangle BDE'| = |\sphericalangle BDE| = 24^\circ$. Preto

$$|\sphericalangle BTE| = |\sphericalangle TDE| + |\sphericalangle TED| = 24^\circ + 36^\circ = 60^\circ.$$

Podobne zo symetrie $|\sphericalangle BTE'| = |\sphericalangle BTE| = 60^\circ$, čiže $|\sphericalangle E'TD'| = 180^\circ - (|\sphericalangle BTE| + |\sphericalangle BTE'|) = 60^\circ$. Teda TE' je osou uhla $BT D'$. Keďže DD' je kolmé ja CE , platí $|\sphericalangle EDD'| = 72^\circ$. Preto $|\sphericalangle D'DE'| = |\sphericalangle EDD'| - |\sphericalangle EDE'| = 24^\circ$, teda aj DE' je osou uhla TDD' . Z toho vyplýva, že E' je stredom kružnice pripísanej k strane TD' trojuholníka TDD' . Čiže $|\sphericalangle TD'E'| = |\sphericalangle DD'C|$. Ale

$$|\sphericalangle TD'E'| + |\sphericalangle DD'C| = 180^\circ - |\sphericalangle ED'D| = |\sphericalangle DED'| + |\sphericalangle EDD'| = 36^\circ + 72^\circ = 108^\circ.$$

Takže $|\sphericalangle DD'C| = 54^\circ$; a tiež $|\sphericalangle CDD'|$. Potom z trojuholníka CDD' dostávame $|\sphericalangle DCD'| = 72^\circ$. Preto $\beta = 72^\circ$ a $\gamma = 12^\circ$.

2.7 (*Pavol Černý*) Keďže $12^2 = 144$, vieme, že dĺžka najdlhšieho bloku je aspoň 2. Preto skúsme nájsť blok dĺžky aspoň 3.

Druhé mocniny dávajú po delení ôsmimi zvyšky 0, 1 alebo 4:

$$(2k)^2 = 4k^2 \equiv 0, 4 \pmod{8} \quad \text{a} \quad (2k+1)^2 = 4k(k+1) + 1 \equiv 1 \pmod{8}.$$

Preto ak využijeme skutočnosť, že 8 delí 1000, môžeme vylúčiť prípady 111, 222, 333, 555, 666 a 999. Ostali len bloky 444, 777 a 888. Avšak žiaden štvorec celého čísla nekončí číslicou 7 ani 8, a preto je jediným možným kandidátom blok 444. Skúšaním nahliadneme, že $38^2 = 1444$, čiže blok dĺžky 3 existuje práve jeden.

Teraz ešte ukážeme, že blok dĺžky 4 už neexistuje. Z predchádzajúceho je zrejmé, že jediným kandidátom je blok 4444. Zrejme musí ísť o štvorec párneho čísla, a preto sú prípustné len dve možnosti zvyškov po delení 16-timi:

$$(4k)^2 = 16k^2 \equiv 0 \pmod{16} \quad \text{a} \quad (4k+2)^2 = 16k(k+1) + 4 \equiv 4 \pmod{16}.$$

Avšak číslo 10 000 je deliteľné 16-timi a číslo 4444 dáva po delení zvyšok 12. Preto nemôže mať žiaden štvorec tvar $10\,000k + 4444$.

Najdlhší možný blok má dĺžku 3 a je to blok 444.

TRETIA SÉRIA

3.1 Predpokladajme, že čísla a, b sú riešením zadanej rovnice. Bez ujmy na všeobecnosti predpokladajme, že $a \geq b$. Uvedomme si platnosť vzťahu $\lfloor x+1 \rfloor > x \geq \lfloor x \rfloor > x-1$. Jeho použitím dostávame nerovnosť

$$\frac{a^2}{b} + \frac{b^2}{a} \geq \left\lfloor \frac{a^2}{b} \right\rfloor + \left\lfloor \frac{b^2}{a} \right\rfloor = \left\lfloor \frac{a^2 + b^2}{ab} \right\rfloor + ab > \frac{a^2 + b^2}{ab} + ab - 1.$$

Po prenasobení nerovnosti číslom ab a následných úpravách máme:

$$\begin{aligned} a^3 + b^3 &> a^2 + b^2 + a^2b^2 - ab, \\ (a^2 - b)(a - b^2) &= a^3 + b^3 - a^2b^2 - ab > a^2 - 2ab + b^2 = (a - b)^2 \geq 0. \end{aligned}$$

Nakoľko $a \geq b$, platí $a^2 - b \geq 0$. Súčin čísel $(a^2 - b)$, $(a - b^2)$ môže byť kladný iba vtedy, ak je kladné $(a - b^2)$. Teda $a > b^2$.

Fakt, že číslo a je väčšie ako b^2 znamená aj to, že $\left\lfloor \frac{b^2}{a} \right\rfloor = 0$. Čísla a, b preto musia vyhovovať aj rovnici

$$\left\lfloor \frac{a^2}{b} \right\rfloor = \left\lfloor \frac{a^2 + b^2}{ab} \right\rfloor + ab.$$

Opätovným použitím vzťahov medzi celými časťami čísel dostaneme nerovnosť :

$$\frac{a^2}{b} - 1 < \left\lfloor \frac{a^2}{b} \right\rfloor = \left\lfloor \frac{a^2 + b^2}{ab} \right\rfloor + ab \leq \frac{a^2 + b^2}{ab} + ab.$$

Prenásobením ab dostaneme nerovnosť $a^3 < a^2b^2 + a^2 + b^2 + ab$.

Ak by bolo $a \geq b^2 + 2$, platila by aj nasledujúca nerovnosť

$$a^3 \geq a^2(b^2 + 2) \geq a^2b^2 + a^2 + a(b^2 + 2) > a^2b^2 + a^2 + a + ab^2 > a^2b^2 + a^2 + b^2 + ab$$

Lenže táto nerovnosť je v rozpore s už dokázanou nerovnosťou, teda predpoklad $a \geq b^2 + 2$ je nesprávny. Nakoniec sme dostali pre riešenie (a, b) podmienku $b^2 + 2 > a > b^2$, ktorej vyhovujú iba dvojice tvaru $(b^2 + 1, b)$.

Lahko sa presvedčíme, že spomínané dvojice riešia zadanú rovnicu (pre $b \geq 2$)

$$\left\lfloor \frac{b^4 + 2b^2 + 1}{b} \right\rfloor + \left\lfloor \frac{b^2}{b^2 + 1} \right\rfloor = b^3 + 2b = \left\lfloor \frac{b^2(b^2 + 1) + (2b^2 + 1)}{(b^2 + 1)b} \right\rfloor + (b^2 + 1)b,$$

a pre $b = 1$ dostávame

$$\left\lfloor \frac{4}{1} \right\rfloor + \left\lfloor \frac{1}{2} \right\rfloor = 4 = \left\lfloor \frac{4 + 1}{2} \right\rfloor + 2.$$

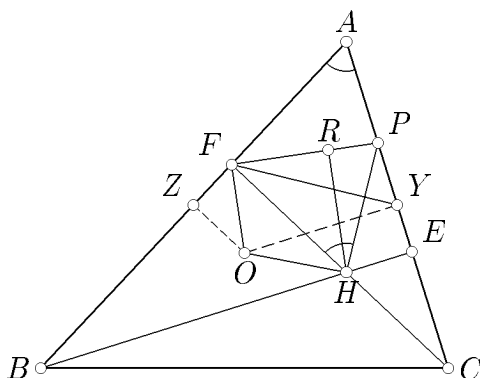
Riešením zadanej rovnice sú usporiadané dvojice čísel v tvare $(n^2 + 1, n)$ a $(n, n^2 + 1)$.

3.2 Označme uhly trojuholníka ako obyčajne α, β, γ . Ďalej si všimnime, že bod O leží vnútri trojuholníka ABC , pretože je ostrouhlý a $|\sphericalangle OCB| = 90^\circ - \alpha < 90^\circ - \beta = |\sphericalangle FCB|$ (pretože $|BC| > |CA|$). To znamená, že O leží vnútri trojuholníka BCF a kolmica na OF prechádzajúca bodom F pretína úsečku AC . To znamená, že P leží na strane AC .

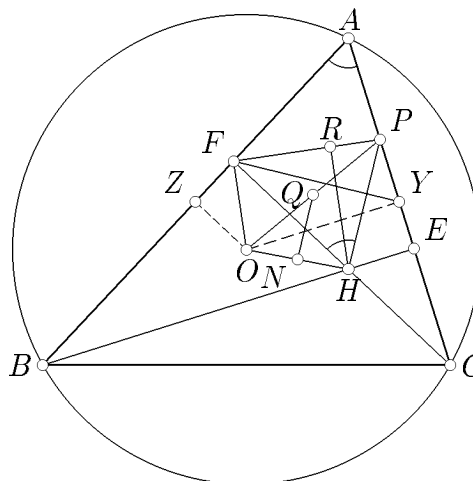
Nech E je päta výšky BH a Y a Z po rade stredy strán AC a AB (obr. 17). Nech R je päta kolmice z bodu H na PF . Teda $HR \perp FP$. Potom Y je stred kružnice opísanej trojuholníku AFC a $|\sphericalangle FYA| = 2|\sphericalangle ACF| = 2(90^\circ - \alpha) = 180^\circ - 2\alpha$. Štvoruholník $OPFY$

je tetivový ($|\sphericalangle PFO| = |\sphericalangle OYP| = 90^\circ$), čo dáva $|\sphericalangle OPF| = |\sphericalangle OYF| = 2\alpha - 90^\circ$. Z podobnosti pravouhlých trojuholníkov OZF a HRF máme

$$\frac{|OZ|}{|OF|} = \frac{|HR|}{|HF|}. \quad (1)$$



Obr. 17



Obr. 18

Teraz určíme vzdialenosť OZ z pravouhlého trojuholníka OZA . Keďže z vety o streďovom a obvodovom uhle vyplýva $|\sphericalangle ZOA| = \gamma$, potom $|OZ| = \frac{|AB|}{2 \operatorname{tg} \gamma}$. Teraz zrejme z trojuholníka BEC máme $|EC| = |BC| \cos \gamma$. No a keďže $|\sphericalangle HCE| = 90^\circ - \alpha$, potom z trojuholníka HCE pre $|HC|$ platí: $|HC| = \frac{|EC|}{\cos(90^\circ - \alpha)} = \frac{|BC| \cos \gamma}{\sin \alpha}$. Zo sínusovej vety ďalej

$$\frac{|BC|}{|AB|} = \frac{\sin \alpha}{\sin \gamma},$$

a preto

$$|HC| = |BC| \cdot \frac{\cos \gamma}{\sin \alpha} = |AB| \cdot \frac{\cos \gamma}{\sin \gamma} = |AB| \cdot \cot \gamma = 2|OZ|. \quad (2)$$

Podobne sa ukáže, že $|HB| = 2|OY|$. Keďže z podobnosti trojuholníkov BHF a CHE vyplýva

$$\frac{|HB|}{|HC|} = \frac{|HF|}{|HE|},$$

dosadením (2) do (1) dostávame

$$|HR| \cdot |OF| = |HF| \cdot |OZ| = \frac{1}{2} |HF| \cdot |HC| = \frac{1}{2} |HE| \cdot |HB| = |HE| \cdot |OY|,$$

teda

$$\frac{|HR|}{|HE|} = \frac{|OY|}{|OF|}.$$

Potom $|\sphericalangle FOY| = |\sphericalangle RHE|$, z toho vyplýva, že trojuholníky RHE a YOF sú podobné, a teda $|\sphericalangle HER| = |\sphericalangle OFY|$. Z tetivových štvoruholníkov $HEPR$ a $OYPF$ dostávame $|\sphericalangle HPR| = |\sphericalangle OPY|$. Teda $|\sphericalangle HPE| = |\sphericalangle OPF| = 2\alpha - 90^\circ$. Napokon z trojuholníka HPC máme $|\sphericalangle FHP| = (2\alpha - 90^\circ) + (90^\circ - \alpha) = \alpha$.

Iné riešenie. Dôkaz, že P leží na AC sme urobili v predchádzajúcom riešení. Nech Y je stred strany AC . Pretože trojuholník AFC je pravouhlý s pravým uhlom pri vrchole F , stredom kružnice jemu opísanej je bod Y . Odtiaľ $|\sphericalangle YFC| = 90^\circ - \alpha$. Takže aby sme ukázali, že $|\sphericalangle FHP| = |\sphericalangle BAC|$, stačí dokázať, že $FY \perp HP$.

Pretože $|\sphericalangle OFP| = |\sphericalangle OYP| = 90^\circ$, body F, P, Y a O ležia na kružnici. Nech je to kružnica k_1 so stredom Q , ktorý je aj stredom úsečky OP . V trojuholníku ABC uvažujme kružnicu deviatich bodov (*Feuerbachovu kružnicu*) k_2 , ktorá prechádza bodmi F a Y a jej stred N je aj stredom OH (obr.18). (To vieme z vlastností *Eulerovej priamky*: body O, H, N spolu s ťažiskom T trojuholníka ABC ležia na jednej priamke v poradí O, T, N a H a známe sú aj pomery vzdialeností medzi nimi.) Kružnice k_1 a k_2 prechádzajú bodmi F a Y , ich spoločnou chordálou je priamka FY . To znamená, že FY je kolmá na NQ (spojnicu stredov týchto dvoch kružníc). Ale NQ je strednou priečkou, a teda je rovnobežná s HP . Odtiaľ dostávame $FY \perp HP$, čo sme chceli dokázať.

3.3 Pre potreby tohto riešenia označme $a_{n+1} = a_1$ a $a_{n+2} = a_2$. Nech

$$A_i = \frac{a_i^2 + a_{i+1}^2 - a_{i+2}^2}{a_i + a_{i+1} - a_{i+2}} = a_i + a_{i+1} + a_{i+2} - \frac{2a_i a_{i+1}}{a_i + a_{i+1} - a_{i+2}}.$$

Keďže $(a_i - 2)(a_{i+1} - 2) \geq 0$, platí aj $-2a_i a_{i+1} \leq -4(a_i + a_{i+1} - 2)$ a

$$A_i \leq a_i + a_{i+1} + a_{i+2} - 4 \left(1 + \frac{a_{i+2} - 2}{a_i + a_{i+1} - a_{i+2}} \right).$$

Pretože $1 = 2 + 2 - 3 \leq a_i + a_{i+1} - a_{i+2} \leq 3 + 3 - 2 = 4$, platí:

$$A_i \leq a_i + a_{i+1} + a_{i+2} - 4 \left(1 + \frac{a_{i+2} - 2}{4} \right) = a_i + a_{i+1} - 2.$$

Sčítaním tejto nerovnosti pre $i = 1, 2, \dots, n$ dostávame

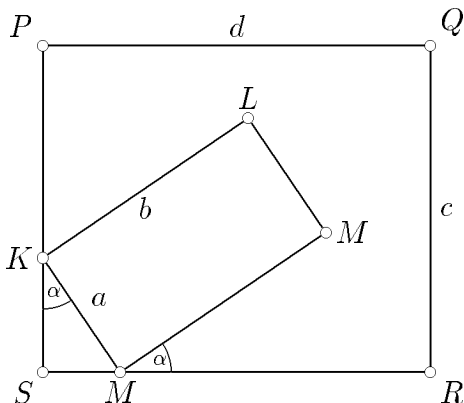
$$\sum_{i=1}^n A_i \leq 2s - 2n.$$

3.4 (*Peter Lupeň*) Evidentne obdĺžnik so stranami $\{a, b\}$ sa dá do obdĺžnika so stranami $\{c, d\}$ umiestniť práve vtedy, keď existuje $\alpha \in (0, \frac{1}{2}\pi)$ také, že platí (obr.19):

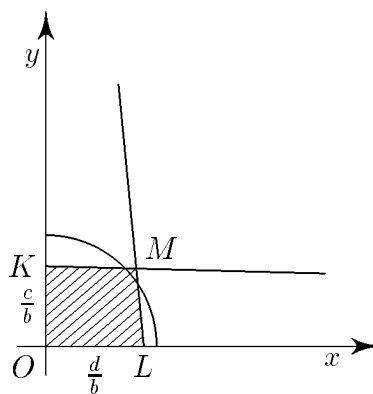
$$\begin{aligned} a \cdot \cos \alpha + b \cdot \sin \alpha &\leq c, \\ a \cdot \sin \alpha + b \cdot \cos \alpha &\leq d. \end{aligned}$$

Položme $x = \cos \alpha$, $y = \sin \alpha$. Potom dostávame ekvivalentné podmienky:

$$x \geq 0, \quad y \geq 0, \quad ax + by \leq c, \quad bx + ay \leq d, \quad x^2 + y^2 = 1.$$



Obr. 19



Obr. 20

Táto sústava sa dá elegantne riešiť, keď si ju geometricky interpretujeme (obr. 20). Prvým štyrom nerovniciam vyhovujú body štvoruholníka $KMLO$ (kde $K = [0, \frac{c}{b}]$, $M = [\frac{bd-ac}{b^2-a^2}, \frac{bc-ad}{b^2-a^2}]$, $L = [\frac{d}{b}, 0]$, $O = [0, 0]$). Aby sa dal prvý obdĺžnik vpísať do druhého, musí ešte platiť aj piata podmienka ($x^2 + y^2 = 1$), čiže musí existovať neprázdny prienik medzi štvoruholníkom $KMLO$ a štvrtkružnicou. Pretože $\frac{c}{b} < 1$, $\frac{d}{b} < 1$, prienik existuje práve vtedy, keď bod M neleží vnútri kruhu, čiže práve vtedy, keď

$$\left(\frac{bd-ac}{b^2-a^2}\right)^2 + \left(\frac{bc-ad}{b^2-a^2}\right)^2 \geq 1,$$

čo nastáva práve vtedy, keď

$$(bd-ac)^2 + (bc-ad)^2 \geq (b^2-a^2)^2, \quad \text{čo bolo treba dokázať.}$$

3.5 (Viera Růžičková) Pre paritu štyroch čísel na začiatku máme len niekoľko možností (stavy, ktoré dostaneme otočením sú zrejme identické): všetky štyri čísla párne $PPPP$, jedno z čísel nepárne $NPPP$, dve susedné čísla nepárne $NNPP$, dve nesusedné čísla nepárne $NPNP$, tri čísla nepárne $NNNP$ a napokon všetky čísla nepárne $NNNN$.

Je zrejmé, že ak sú všetky štyri čísla na kružnici v nejakom okamihu párne, ostávajú párnymi aj po každom ďalšom kroku. Jednoducho dokážeme, že každý možný stav sa po pár krokoch zmení na stav $PPPP$.

Stav $PPPP$ je triviálny. Stav $NPPP$ sa postupne transformuje na $NNPP$, potom na $NPNP$, a ten na $NNNN$, ktorý po jednom kroku prejde na $PPPP$. Týmto sme teda hneď vyriešili aj stavy $NNPP$, $NPNP$ a $NNNN$.

Zostávajúci stav – $NNNP$ – sa po jednom kroku transformuje na $NNPP$, ktorý, ako už vieme, sa po troch ďalších krokoch zmení na stav $PPPP$.

Takto sme dokázali, že v každom prípade budú po 1997 krokoch na kružnici len párne čísla. Potom sú ale všetky tri výrazy $|bc - ad|$, $|ac - bd|$ a $|ab - cd|$ deliteľné štyrmi, a teda dokonca žiaden z nich nemôže byť prvočíslom.

Iné riešenie. (*Peter Kozák a Peter a Pavol Novotný*)

Ak sú na začiatku na kružnici napísané čísla a, b, c a d , tak po prvom kroku dostaneme po rade čísla $a - b, b - c, c - d$ a $d - a$. Vo výpočte pokračujeme ďalej až po štvrtý krok:

$$\begin{array}{cccc}
 a & b & c & d \\
 a - b & b - c & c - d & d - a \\
 a - 2b + c & b - 2c + d & c - 2d + a & d - 2a + b \\
 a - 3b + 3c - d & b - 3c + 3d - a & c - 3d + 3a - b & d - 3a + 3b - c \\
 2a - 4b + 6c - 4d & 2b - 4c + 6d - 4a & 2c - 4d + 6a - 4b & 2d - 4a + 6b - 4c
 \end{array}$$

Opäť si všimneme, že všetky štyri čísla sú už po štyroch krokoch párne, čiže použitím rovnakých úvah ako v predchádzajúcom riešení zistíme, že všetky štyri skúmané výrazy musia byť deliteľné štyrmi.

3.6 Dosadením čísel $x = 1, -1, \frac{1}{2}, -\frac{1}{2}$ do polynómu dostaneme podmienky

$$|a + b + c + d| \leq 1, \quad (1)$$

$$|-a + b - c + d| \leq 1, \quad (2)$$

$$|a + 2b + 4c + 8d| \leq 8, \quad (3)$$

$$|-a + 2b - 4c + 8d| \leq 8. \quad (4)$$

Tieto nerovnice si môžeme predstaviť ako hraničné nadroviny štvorrozmerného rovno-bežnostenu. Maximum súčtu $|a| + |b| + |c| + |d|$ sa zrejme nadobúda v jeho vrcholoch. Preto stačí nájsť jeho vrcholy, čo znamená riešiť 16 sústav lineárnych rovníc so štyrmi neznámymi. Ich vyriešením a overením podmienky $|a| + |b| + |c| + |d| \leq 7$ pre vrcholy, overíme platnosť zadaného tvrdenia.

Možno postupovať aj inak. Z prvých dvoch nerovností dostávame

$$2|b + d| \leq |(b + d) + (a + c)| + |(b + d) - (a + c)| \leq 2 \quad \Rightarrow \quad |b + d| \leq 1.$$

Ak $b + d \geq 0$, potom z (2), $a + c \geq (b + d) - 1$, na druhej strane z (1) $a + c \leq 1 - (b + d)$. Preto

$$|b + d| + |a + c| \leq |b + d| + |1 - (b + d)| = (b + d) + 1 - (b + d) = 1. \quad (5)$$

Rovnako aj v prípade $b + d \leq 0$ dostaneme rovnakú nerovnosť.

Potom

$$|a| - |c| = |a| - |-c| \leq |a + c|,$$

z čoho pomocou (5)

$$|b + d| + |a| - |c| \leq |b + d| + |a + c| \leq 1. \quad (6)$$

Podobne z nerovností (3) a (4) dostaneme

$$2|b + 4d| + 4|c| - |a| = |2b + 8d| + |4c| - |a| \leq 8. \quad (7)$$

Prenásobením (6) číslom 5 a (7) číslom 2 a sčítaním nerovností dostávame

$$3|a| + 3|c| + 4|b + 4d| + 5|b + d| \leq 21. \quad (8)$$

Vzhľadom na dokazovanú nerovnosť nám teda stačí dokázať už len nerovnosť

$$3|b| + 3|d| \leq 4|b + 4d| + 5|b + d|, \quad (9)$$

lebo spojením (8) a (9) dostávame

$$3(|a| + |b| + |c| + |d|) \leq 21.$$

Nerovnosť (9) platí pre $d = 0$ triviálne. Ak $d \neq 0$, položíme $x = \frac{b}{d}$. Stačí nám teraz dokázať, že funkcia

$$f(x) = 4|x + 4| + 5|x + 1| - 3|x| - 3$$

je nezáporná pre všetky reálne čísla x . To sa však ľahko ukáže, stačí uvažovať $f(x)$ na štyroch disjunktných intervaloch $(-\infty, -4)$, $(-4, -1)$, $(-1, 0)$ a $(0, \infty)$:

$$-\infty < x \leq -4: \quad f(x) = -5(x + 1) - 4(x + 4) + 3x - 3 = -6(x + 4) \geq 0;$$

$$-4 < x \leq -1: \quad f(x) = -5(x + 1) + 4(x + 4) + 3x - 3 = 2(x + 4) \geq 0;$$

$$-1 < x \leq 0: \quad f(x) = 5(x + 1) + 4(x + 4) + 3x - 3 = 12x + 18 \geq 0;$$

$$0 < x < \infty: \quad f(x) = 5(x + 1) + 4(x + 4) - 3x - 3 = 6x + 18 \geq 0.$$

Tým je zadaná nerovnosť dokázaná.

3.7 (Viera Růžičková) Ak je v podmnožine \mathcal{S} súčet najmenších n prvkov väčší alebo rovný m , tak to platí pre všetky jej n -prvkové podmnožiny a naopak, ak to platí pre všetky tieto podmnožiny, platí to aj pre n najmenších prvkov.

Ak \mathcal{S} obsahuje z prvkov tak, že spĺňa podmienku, aj podmnožina množiny $\{1, \dots, k\}$ obsahujúca jej z najväčších prvkov spĺňa danú podmienku, lebo súčet najmenších n prvkov nemá menší. Stačí nám teda uvažovať len takéto podmnožiny. Potom má \mathcal{S} najväčšiu mohutnosť práve vtedy, keď $\mathcal{S} = \{k, k - 1, k - 2, \dots, x\}$, pričom platí

$$m \leq x + (x + 1) + (x + 2) + \dots + (x + n - 1) = \frac{1}{2}(2x + n - 1)n,$$

$$m > (x - 1) + x + (x + 1) + \dots + (x + n - 2) = \frac{1}{2}(2x + n - 3)n.$$

Teda \mathcal{S} má najväčšiu mohutnosť práve vtedy, keď

$$\frac{1}{2} \left(\frac{2m}{n} + 1 - n \right) + 1 > x \geq \frac{1}{2} \left(\frac{2m}{n} + 1 - n \right).$$

(Mohutnosť \mathcal{S} je $k - x + 1$, čo je najväčšie práve vtedy, keď x je najmenšie.) Keď mohutnosť \mathcal{S} označíme h , potom $h = k - x + 1$ je najväčšie práve vtedy, keď

$$k - \frac{1}{2} \left(\frac{2m}{n} + 1 - n \right) < h \leq k - \frac{1}{2} \left(\frac{2m}{n} + 1 - n \right) + 1,$$

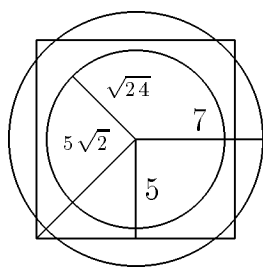
teda práve vtedy, keď

$$h = \left\lfloor k - \frac{1}{2} \left(\frac{2m}{n} + 1 - n \right) + 1 \right\rfloor.$$

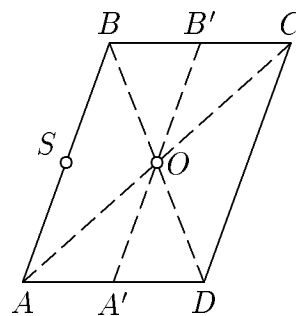
ŠTVRTÁ SÉRIA

4.1 (*Juraj Földes*)

a) Označme v_1 a v_2 po rade výšky (vzdialenosti od podlahy – jednej steny kocky) bodov A a B . Vezmime si rozdiel výšok múch. Na začiatku je $v_1 - v_2$ a na konci je $v_2 - v_1$. To znamená, že v niektorom momente bude rozdiel ich výšok rovný nule (pretože muchy sa pohybujú spojitاً, teda aj ich výšky a aj rozdiel ich výšok sa mení spojitاً), čo znamená, že v tom momente budú v rovnakej výške, a teda budú ležať v jednej rovine rovnobežnej s podlahou. Budú ležať vo štvorci (myslíme tým vnútri alebo na stranách) so stranou 10 m (obr. 21). Ale maximálna vzdialenosť dvoch bodov vo štvorci je dĺžka jeho uhlopriečky, čo je v našom prípade $10\sqrt{2} < 15$. Teda v tomto momente bude aj vzdialenosť múch menšia ako 15 m.



Obr. 21



Obr. 22

b) Ak je počiatočná vzdialenosť múch $|AB| = 14$ m, ukážeme, že odpoveď je áno (inak je zrejme nie). Nech teda $|AB| = 14$ m, Označme S stred úsečky AB . Najprv posunieme úsečku AB v posunutí, v ktorom sa bod S zobrazí na stred kocky O . Dostaneme tak úsečku $A'B'$ ($|A'B'| = 14$ m). Ukážeme že úsečky AA' , BB' ležia v kocke (myslí sa tým na povrchu alebo vnútri), obr. 22. Kocka je teleso súmerné podľa stredu O , teda ak A, B ležia v kocke, potom aj body C, D súmerne združené s A, B podľa stredu O

ležia v kocke. Ale zrejme A' je stred AD a B' je stred BC . Z konvexnosti kocky potom vyplýva, že úsečky AA' , BB' ležia v kocke.

Nech muchy lietajú stále navzájom rovnakou rýchlosťou. Necháme ich najprv preletieť do A' , B' . Teraz si vezmeme guľovú plochu $\mathcal{G}(O; 7 \text{ m})$. Označme α jej prienik s kockou (potom $A', B' \in \alpha$). Zrejme \mathcal{G} nepretína žiadnu hranu kocky (priemet \mathcal{G} do roviny steny kocky je znázornený na obr. 21, kde väčšia kružnica znázorňuje „obrys“ guľovej plochy \mathcal{G} , a menšia jej prienik so stenou kocky), a teda α je „súvislá“ množina bodov, čiže po α sa dá dostať z každého jej bodu do každého iného jej bodu. Teda aj z A' do B' . Necháme touto cestou letieť muchu \mathcal{F} a mucha \mathcal{M} bude letieť stredovo súmerne podľa O s muchou \mathcal{F} (bude letieť po α , lebo α je zrejme stredovo súmerná podľa stredu O a ich vzdialenosť bude zrejme vždy práve 14 m). No a potom už muchy preletia do bodov B a A , čo je zrejme možné (pozri začiatok).

4.2 Najprv ukážeme, že polynóm $P(x) - Q(x)$ má aspoň jeden reálny koreň, čiže existuje také reálne číslo t , pre ktoré $P(t) = Q(t)$.

Nech u a v sú po rade korene $P(x)$ a $Q(x)$, čiže $P(u) = 0$ a $Q(v) = 0$. Ak $u = v$ je vyššie uvedené tvrdenie zrejmé. Preto nech $u \neq v$, bez ujmy na všeobecnosti $u < v$. Pre

$$R(x) = P^2(x) - Q^2(x)$$

platí $R(u) \leq 0$ a $R(v) \geq 0$. Zo spojitosti polynómu $R(x)$ vyplýva existencia takého s , $s \in \langle u, v \rangle$, pre ktoré platí $P^2(s) = Q^2(s)$. Označme $t = 1 + s + Q^2(s)$. Zo zadania platí

$$P(t) = P(1 + s + Q^2(s)) = Q(1 + s + P^2(s)) = Q(1 + s + Q^2(s)) = Q(t).$$

Takto sme dokázali existenciu aspoň jedného spoločného bodu polynómov $P(x)$ a $Q(x)$. Ďalej označme

$$t_1 = t \quad \text{a} \quad t_{i+1} = 1 + t_i + Q^2(t_i).$$

Matematickou indukciou podľa i dokážeme, že pre každé $i \in \mathbb{N}$ platí $P(t_i) = Q(t_i)$.

1° Pre $i = 1$ sme tvrdenie už dokázali.

2° Predpokladajme, že tvrdenie platí pre $i = k$. Potom

$$P(t_{k+1}) = P(1 + t_k + Q^2(t_k)) = Q(1 + t_k + P^2(t_k)) = Q(1 + t_k + Q^2(t_k)) = Q(t_{k+1}).$$

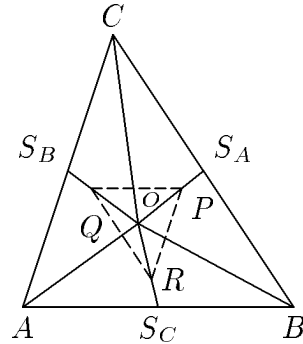
Preto má polynóm $P(x) - Q(x)$ nulovú hodnotu pre $x = t_i$, $i = 1, 2, \dots$. Poľahky nahliadneme, že platí $t_{i+1} = 1 + t_i + Q^2(t_i) > t_i$, a preto má polynóm $P(x) - Q(x)$ nekonečne veľa koreňov, čiže je nulový. Teda $P(x) \equiv Q(x)$.

4.3 Označme S_C, S_B, S_A stredy strán AB, BC, CA a T ťažisko trojuholníka ABC (obr. 23). Zrejme trojuholník $S_A S_B S_C$ je obrazom trojuholníka ABC v rovnoľahlosti so stredom T a koeficientom $-\frac{1}{2}$ (zapisujeme $\mathcal{H}_{(T, -\frac{1}{2})}(\triangle ABC \rightarrow \triangle S_A S_B S_C)$). Keďže

$$\frac{|OP|}{|OS_A|} = \frac{|OQ|}{|OS_B|} = \frac{|OR|}{|OS_C|} = \frac{2}{3},$$

potom taktiež $\mathcal{H}_{(O, \frac{2}{3})}(\triangle S_A S_B S_C) \rightarrow \triangle PQR$.

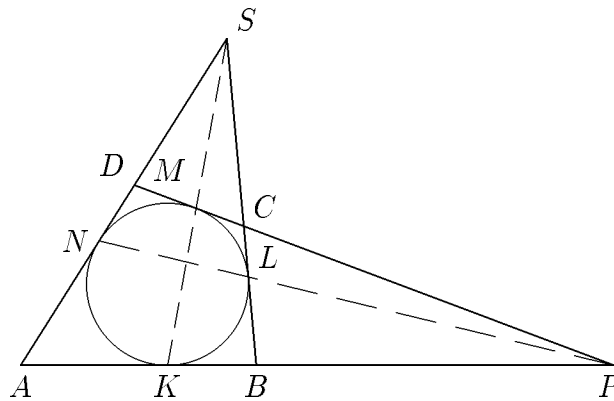
Ak označíme $\mathcal{Z} = \mathcal{H}_{(T, -\frac{1}{2})} \circ \mathcal{H}_{(O, \frac{2}{3})}$ zloženie rovnoľahlostí (pričom ako prvá sa vykoná tá vľavo), tak $\mathcal{Z}(\triangle ABC \rightarrow \triangle PQR)$. Nech S je stred vpísanej kružnice trojuholníku PQR . Potom platí $\mathcal{Z}(O \rightarrow S)$. Vezmime teraz bod O' taký, že $\mathcal{H}_{(T, -\frac{1}{2})}(O \rightarrow O')$. Zjavne O' leží na polpriamke \overrightarrow{OT} a navyše $|O'O| = \frac{3}{2}|TO|$, čiže $|TO| = \frac{2}{3}|O'O|$. Potom však $\mathcal{H}_{(O, \frac{2}{3})}(O' \rightarrow T)$, z čoho $T \equiv S$. Tým je tvrdenie dokázané.



Obr. 23

Poznámka. V špeciálnom prípade ak $T \equiv O$, sú body O a O' v oboch rovnoľahlostiach samodružné, čiže aj $S \equiv O \equiv T$.

4.4 (Vladimír Marko) Označme $a = |AK| = |AN|$, $b = |BK| = |BL|$, $c = |CL| = |CM|$, $d = |DM| = |DN|$, $s = |SN| = |SL|$, $p = |PK| = |PM|$. Použitím Menelaovej vety pre trojuholník APD a kolineárne body K, M, S (ležia na \overleftrightarrow{AP} , \overleftrightarrow{PD} , \overleftrightarrow{AD}) dostávame (obr. 24):



Obr. 24

$$\frac{|KA|}{|KP|} \cdot \frac{|MP|}{|MD|} \cdot \frac{|SD|}{|SA|} = 1 \quad \implies \quad \frac{a}{p} \cdot \frac{p}{d} \cdot \frac{s-d}{s+a} = 1. \quad (1)$$

Použitím Menelaovej vety pre trojuholník BPC a kolineárne body K, M, S (ležia na \overleftrightarrow{BP} , \overleftrightarrow{PC} , \overleftrightarrow{BC}) dostávame:

$$\frac{|KB|}{|KP|} \cdot \frac{|MP|}{|MC|} \cdot \frac{|SC|}{|SB|} = 1 \quad \implies \quad \frac{b}{p} \cdot \frac{p}{c} \cdot \frac{s-c}{s+b} = 1. \quad (2)$$

Označme P' priesečník \overleftrightarrow{NL} a \overleftrightarrow{AB} , P'' priesečník \overleftrightarrow{NL} a \overleftrightarrow{CD} , nech $p' = |P'K|$, $p'' = |P''M|$. Použitím *Menelaovej vety* pre trojuholník ABS a body N, L, P' máme:

$$\frac{|NA|}{|NS|} \cdot \frac{|LS|}{|LB|} \cdot \frac{|P'B|}{|P'A|} = 1 \quad \implies \quad \frac{a}{s} \cdot \frac{s}{b} \cdot \frac{p' - b}{p' + a} = 1. \quad (3)$$

Použitím *Menelaovej vety* pre trojuholník CDS a body N, L, P'' máme:

$$\frac{|ND|}{|NS|} \cdot \frac{|LS|}{|LC|} \cdot \frac{|P''C|}{|P''D|} = 1 \quad \implies \quad \frac{d}{s} \cdot \frac{s}{c} \cdot \frac{p'' - c}{p'' + d} = 1. \quad (4)$$

Potom z (1) a (2) dostaneme $\frac{2ad}{a-d} = s = \frac{2bc}{b-c}$, čo po úpravách dáva $\frac{2cd}{d-c} = \frac{2ab}{a-b}$.

Z (3) a (4) dostávame $p' = \frac{2ab}{a-b}$, $p'' = \frac{2cd}{d-c}$. Vidíme, že $p' = p''$, a teda aj $P'' \equiv P' \equiv P$, čo sme mali dokázať.

Poznámka. Treba ešte ukázať, že $a-d \neq 0$ a $d-c \neq 0$ (týmito výrazmi sme delili). Lenže ak by bolo $a = d$, potom $|AN| = |DN|$ a $KM \parallel AD$ a neexistoval by bod S . Rovnako v prípade $b = c$.

4.5 (*Miroslav Dudík*) Tvrdenie dokážeme sporom. Nech taká stratégia neexistuje. Označme \mathcal{D} danú množinu úsečiek. Existencia vyhrávajúcej stratégie formálne zapísaná je ekvivalentná s tvrdením, že

$$\begin{aligned} \forall \mathcal{D} : \exists a \in \mathcal{D} : \forall a' \in \mathcal{D} \setminus \{a\} : \exists b \in \mathcal{D} \setminus \{a, a'\} : \forall b' \in \mathcal{D} \setminus \{a, a', b\} : \\ \exists c \in \mathcal{D} \setminus \{a, a', b, b'\} : \forall c' \in \mathcal{D} \setminus \{a, a', b, b', c\} : S(a, b, c) \geq S(a', b', c'), \end{aligned}$$

kde $S(a, b, c)$ je obsah trojuholníka so stranami a, b a c . Preto je negácia tohto tvrdenia nasledujúca:

$$\begin{aligned} \exists \mathcal{D} : \forall a \in \mathcal{D} : \exists a' \in \mathcal{D} \setminus \{a\} : \forall b \in \mathcal{D} \setminus \{a, a'\} : \exists b' \in \mathcal{D} \setminus \{a, a', b\} : \\ \forall c \in \mathcal{D} \setminus \{a, a', b, b'\} : \exists c' \in \mathcal{D} \setminus \{a, a', b, b', c\} : S(a, b, c) < S(a', b', c'), \end{aligned}$$

teda existencia vyhrávajúcej stratégie pre druhého hráča (pre konkrétnu množinu \mathcal{D}).

Označme hráčov po rade \mathcal{H}_1 a \mathcal{H}_2 . Ďalej nech $P(a)$ je úsečka a' , ktorú by si bral v druhom ťahu \mathcal{H}_2 , ak by \mathcal{H}_1 bral v prvom ťahu a , ďalej $P(a, b)$ úsečku b' , ktorú by ťahal v štvrtom ťahu \mathcal{H}_2 , keby boli postupne ťahané $a, P(a), b$, a napokon $P(a, b, c)$ úsečku, ktorú by ťahal v poslednom ťahu \mathcal{H}_2 , keby boli postupne ťahané $a, P(a), b, P(a, b), c$. Zrejme $P(a, b, c)$ je jediný prvok množiny $\mathcal{D} \setminus \{a, b, c, P(a), P(a, b)\}$. K sporu dôjdeme tak, že ukážeme, že \mathcal{H}_1 môže docieľiť koncovú situáciu, v ktorej bude mať úsečky $P(a)$, $P(a, b)$ a $P(a, b, c)$ (pričom podľa predpokladu existuje víťazná stratégia pre \mathcal{H}_2 , kde \mathcal{H}_1 ťahá úsečky a, b a c).

Nech \mathcal{H}_1 ťahá v prvom ťahu $P(u_1)$, pričom u_1 je ľubovoľná úsečka z \mathcal{D} . Potom má \mathcal{H}_2 dve možnosti (podľa svojej víťaznej stratégie):

- ťahá u_1 ; potom nech ďalej \mathcal{H}_1 ťahá $P(u_1, u_2)$, pričom u_2 je ľubovoľné z $\mathcal{D} \setminus \{u_1, P(u_1)\}$.
- ťahá u_2 , $u_2 \in \mathcal{D} \setminus \{u_1, P(u_1)\}$; potom nech \mathcal{H}_1 ťahá $P(u_1, u_2)$, zrejme $P(u_1, u_2) \neq P(u_1)$ a $P(u_1, u_2) \neq u_2$.

Po týchto troch ťahoch má teda prvý hráč $P(u_1)$ a $P(u_1, u_2)$ a \mathcal{H}_2 má u_1 alebo u_2 . Opäť sú dve možnosti:

- \mathcal{H}_2 ťahá u_2 , resp. u_1 ; potom \mathcal{H}_1 ťahá ľubovoľnú úsečku, lebo pre dve zvyšné úsečky u_3, u_4 v \mathcal{D} platí $P(u_1, u_2, u_3) = u_4$ a zároveň $P(u_1, u_2, u_4) = u_3$.
- \mathcal{H}_2 ťahá u_3 , $u_3 \notin \{u_1, u_2\}$; potom \mathcal{H}_1 ťahá $P(u_1, u_2, u_3)$, ktorá iste nebola ešte vytiahnutá (je to tá úsečka, ktorá je rôzna od zvyšnej u_1 príp. u_2).

Na konci hry má teda \mathcal{H}_1 úsečky $P(u_1), P(u_1, u_2)$ a $P(u_1, u_2, u_3)$. Na druhej strane \mathcal{H}_2 má zvyšné úsečky, čiže u_1, u_2, u_3 . Keďže sme však predpokladali, že $S(u_1, u_2, u_3) < S(P(u_1), P(u_1, u_2), P(u_1, u_2, u_3))$, dostávame spor. Preto pre \mathcal{H}_1 vždy existuje vyhrávajúca stratégia.

Iné riešenie. (Pavol Černý)

Hra môže mať $\frac{\binom{6}{3}}{2!} = 10$ rôznych priebehov. Existuje teda 10 neprehrávajúcich trojíc úsečiek. Označme úsečky po rade u_1, \dots, u_6 tak, aby $P(u_1) \geq P(u_i)$, pre $i = 2, \dots, 6$, kde $P(u_i)$ je počet výskytov úsečky u_i v neprehrávajúcich trojiciach. Rozoberieme tri prípady.

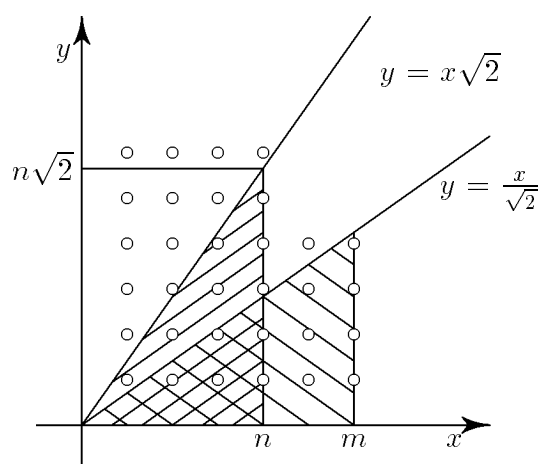
- $P(u_1) \geq 7$;
- $P(u_1) = 6$ a $P'(u_2) \leq 3$ alebo $P(u_1) = 5$ a $P'(u_2) \leq 2$;
- $P(u_1) = 6$ a $P'(u_2) = 4$ alebo $P(u_1) = 5$ a $P'(u_2) = 4$ alebo $P(u_1) = 5$ a $P'(u_2) = 3$.

(u_2 je úsečka, ktorá sa najčastejšie vyskytuje v neprehrávajúcich trojiciach v kombinácii s u_1 a $P'(u_2)$ je počet týchto výskytov.) Keďže spolu s u_1 a u_2 môžu byť v neprehrávajúcich trojiciach nanajvyš štyri iné úsečky, platí $P'(u_2) \leq 4$. Ďalej na 30 miest v neprehrávajúcich trojiciach umiestňujeme 6 úsečiek, preto podľa *Dirichletovho princípu* platí $P(u_1) \geq 5$. Z toho okamžite vidieť, že prípady a), b) a c) predstavujú všetky možnosti.

- $P(u_1) \geq 7$. Vezmime si tie neprehrávajúce trojice, v ktorých je u_1 a nie u_2 . Sú aspoň tri ($7-4=3$). V týchto trojiciach je aspoň šesť miest, na ktoré umiestňujeme štyri zvyšné úsečky, preto podľa *Dirichletovho princípu* existuje úsečka u_3 , ktorá je tam aspoň dvakrát. Nevyhrávajúca stratégia pre prvého hráča je potom:
 - prvý ťah (\mathcal{H}_1) u_1 , potom dva prípady:
 - i) druhý ťah (\mathcal{H}_2) u_2 , potom tretí ťah (\mathcal{H}_1) u_3 a ostanú aspoň dve neprehrávajúce trojice, v ktorých je u_1, u_3 a nie u_2 . Hráč \mathcal{H}_2 môže svojim ťahom zrušiť nanajvyš jednu a ťah \mathcal{H}_1 povedie k druhej.
 - ii) druhý ťah (\mathcal{H}_2) nie je u_2 , napr. u_4 , potom tretí ťah (\mathcal{H}_1) je u_2 a opäť ostali aspoň dve neprehrávajúce trojice obsahujúce u_1 a u_2 a nie u_4 ($P'(u_2) \geq P'(u_3) \geq 2$) a ďalej možno postupovať ako v i).
- b) Možno postupovať podobne ako v a), existuje také u_3 , že $P'(u_3) \geq 2$ a stratégia je rovnaká ako v a).

- c) Najprv prípady $P(u_1) = 6$ a $P'(u_2) = 4$ alebo $P(u_1) = 5$ a $P'(u_2) = 4$. Existujú aspoň 4 nevyhrávajúce trojice obsahujúce u_1 . Tieto nevyhrávajúce trojice už zrejme neobsahujú u_2 (všetky kombinácie u_1 a u_2 – štyri – sú už vo vyhrávajúcich trojiciach). Im odpovedajúce neprehrávajúce trojice u_2 už obsahovať musia, z toho $P(u_2) = 4 + P'(u_2) = 8$, čo je spor, pretože $P(u_1) \geq P(u_2)$, teda takýto prípad nemôže nastať. Pre $P(u_1) = 5$ a $P'(u_2) = 3$ dostaneme obdobný spor.

4.6 (*Peter Svrček*) Dokazovaná identita je ekvivalentná s tým, že dve vyšrafované oblasti na obr.25 obsahujú rovnaký počet mrežových bodov (mrežové body na x -ovej osi nepočítame).



Obr.25

Z geometrickej interpretácie okamžite vychádza, že

$$\sum_{k=0}^n \lfloor k \cdot \sqrt{2} \rfloor = \sum_{\substack{i=1 \\ i < n\sqrt{2}}} \left(n - \left\lfloor \frac{i}{\sqrt{2}} \right\rfloor \right).$$

(Mrežové body v prvej oblasti sčítame nie po stĺpcoch, ale po riadkoch ako doplnok počtu mrežových bodov v obdĺžniku k bodom nad grafom.)

Teraz sa pokúsime stanoviť presnú hranicu, pokiaľ sčítame v sume. Dokážeme, že

$$\sum_{\substack{i=1 \\ i < n\sqrt{2}}} \left(n - \left\lfloor \frac{i}{\sqrt{2}} \right\rfloor \right) = \sum_{i=1}^{m-1} \left(n - \left\lfloor \frac{i}{\sqrt{2}} \right\rfloor \right).$$

Stačí dokázať $m-1 < n\sqrt{2} < m$. Z daného vzťahu $2(n+2)n = (m+2)m$ po pripočítaní dvojky na obe strany dostaneme

$$2(n+1)^2 = 2(n+2)n + 2 = m^2 + 2m + 2, \quad \text{čiže} \quad n = \sqrt{\frac{m(m+2)+2}{2}} - 1.$$

Potom

$$n\sqrt{2} = \sqrt{2} \left(\sqrt{\frac{m(m+2)}{2} + 1} - 1 \right) = \sqrt{(m+1)^2 + 1} - \sqrt{2} < m.$$

Poslednú nerovnosť sme dostali z nerovnosti

$$m^2 + 2m + 2 = (m+1)^2 + 1 < (m + \sqrt{2})^2 = m^2 + 2\sqrt{2}m + 2.$$

Ešte dokážeme, že

$$m - 1 < n\sqrt{2} = \sqrt{2} \left(\sqrt{\frac{m(m+2)}{2} + 1} - 1 \right) = \sqrt{(m+1)^2 + 1} - \sqrt{2}.$$

Platí

$$\begin{aligned} m - 1 + \sqrt{2} &< \sqrt{(m+1)^2 + 1}, \\ m^2 + 2(\sqrt{2} - 1)m + 3 - 2\sqrt{2} &< m^2 + 2m + 2, \end{aligned}$$

pretože $2(\sqrt{2} - 1) < 2$ a $3 - 2\sqrt{2} < 2$ a umocnenie bola ekvivalentná úprava.

Obdobným spôsobom ukážeme, že $n = \left\lfloor \frac{m}{\sqrt{2}} \right\rfloor$, čiže

$$n \leq \frac{m}{\sqrt{2}} < n + 1.$$

Preto $n - \left\lfloor \frac{m}{\sqrt{2}} \right\rfloor = 0$, a teda

$$\sum_{k=0}^n \left\lfloor k \cdot \sqrt{2} \right\rfloor = \sum_{i=1}^{m-1} \left(n - \left\lfloor \frac{i}{\sqrt{2}} \right\rfloor \right) = \sum_{i=1}^m \left(n - \left\lfloor \frac{i}{\sqrt{2}} \right\rfloor \right).$$

Preto je dokazovaná identita ekvivalentná s identitou

$$\sum_{i=1}^m \left\lfloor \frac{i}{\sqrt{2}} \right\rfloor = \sum_{i=1}^m \left(n - \left\lfloor \frac{i}{\sqrt{2}} \right\rfloor \right),$$

čiže

$$\sum_{i=1}^m n = 2 \cdot \sum_{i=1}^m \left\lfloor \frac{i}{\sqrt{2}} \right\rfloor.$$

To však zrejme na základe obr. 25 platí, pretože priamka $y = x\sqrt{2}$ neprechádza žiadnym mrežovým bodom, a preto obsahuje vyšrafovaná časť presne polovicu mrežových bodov vo vnútri obdĺžnika $n \times m$, ktorý obsahuje práve $\sum_{i=1}^m n$ mrežových bodov.

4.7 (*Vladimír Marko*) Dokážeme, že také prirodzené číslo n neexistuje.

Pre $n = 2$ je súčet $n - 1$ čísel len jedno číslo, ktoré zrejme patrí len do tej množiny, v ktorej je, a teda toto n nevyhovuje.

Pre $n \geq 3$ budeme postupovať sporom. Nech sa dá množina \mathbb{N} rozdeliť na n podmnožín spĺňajúcich danú podmienku. Potom medzi týmito množinami (podľa *Dirichletovho princípu*) existuje množina \mathcal{A} , ktorá obsahuje aspoň dve čísla z množiny $\{1, 2, \dots, n + 1\}$, zrejme je rozdiel medzi týmito dvomi číslami nanajvýš n . Rozlíšime dva prípady:

- Existuje taká množina \mathcal{A} , že rozdiel medzi niektorými jej prvkami je menší ako n , označme tieto čísla x a $x + d$, $d < n$. Vezmime teraz ľubovoľnú inú množinu \mathcal{B} a jej ľubovoľný prvok y . Potom

$$y \in \mathcal{B} \quad \Rightarrow \quad y + d \in \mathcal{B} \cup \mathcal{A}. \quad (1)$$

Ak by totiž $y + d \in \mathcal{C}$, $\mathcal{C} \not\subseteq \mathcal{A} \cup \mathcal{B}$, potom by sme vzali súčet S ľubovoľných $n - 3$ čísel z množín iných ako $\mathcal{A}, \mathcal{B}, \mathcal{C}$ (prípadne pre $n = 3$ máme $S = 0$), potom $S + x + (y + d)$ má patriť \mathcal{B} a $S + (x + d) + y$ má patriť \mathcal{C} , čo je spor, lebo $S + (x + d) + y = S + x + (y + d)$.

Teraz vezmime množinu \mathcal{A} tak, že najmenší rozdiel medzi jej dvomi prvkami d je najmenší zo všetkých množín rozkladu množiny \mathbb{N} . Potom čísla $x + 1, x + 2, \dots, x + d - 1$ musia patriť rôznym množinám rôznym od \mathcal{A} . Čísla $x + d + 1, x + d + 2, \dots, x + 2d - 1$ podľa už dokázaného tvrdenia (1) patria buď \mathcal{A} , čo vzhľadom na minimalitu d nie je možné, alebo tej istej množine, ktorá obsahuje číslo o d menšie. Preto dvojice $(x + 1, x + d + 1)$, $(x + 2, x + d + 2)$ až $(x + d - 1, x + 2d - 1)$ patria do jednej množiny. Teraz však zrejme môžeme množinu \mathcal{A}' obsahujúcu dvojicu $(x + 1, x + d + 1)$ zvoliť za počiatočnú a rovnakou úvahou dostaneme, že dvojica prvkov $(x + 1 + (d - 1), x + d + 1 + (d - 1)) = (x + d, x + 2d)$ patrí tej istej množine, čiže aj $x + 2d \in \mathcal{A}$. Dvojicu prvkov $(x + d, x + 2d)$ môžeme teraz zvoliť za $(x', x' + d)$ atď. Postupne dostávame, že v množine \mathcal{A} ležia všetky prvky $x, x + d, x + 2d, \dots, x + kd, \dots$, prvky $x + 1, x + d + 1, x + 2d + 1, \dots, x + kd + 1, \dots$ v ďalšej množine atď.

Tvrdenie (1) zrejme platí aj naopak (ľahko sa obrátia úvahy), čiže

$$y + d \in \mathcal{B} \quad \Rightarrow \quad y \in \mathcal{A} \cup \mathcal{B}$$

a celú predchádzajúcu úvahu možno rozšíriť aj na čísla menšie ako x a dostávame rozklad \mathbb{N} nie na n ale na d , $d < n$ množín, čo je spor.

- Neexistuje množina \mathcal{A} s číslami $x, x + d$, $d < n$. Keďže sme však už dokázali, že $d \leq n$, potom použitím úvah ako v predchádzajúcom prípade dostaneme rozklad množiny \mathbb{N} na zvyškové triedy modulo n , čiže v jednotlivých množinách budú čísla dávajúce ten istý zvyšok po delení číslom n . Potom ak sčítame po jednom čísle z každej množiny a súčet označíme S , zrejme

$$S \equiv \sum_{i=1}^n i \pmod{n}.$$

Číslo $S - 1$ musí patriť zvyškovej triede obsahujúcej 1, čiže $S - 1 \equiv 1 \pmod{n}$ a podobne $S - 2 \equiv 2 \pmod{n}$. Tak však dostávame, že zároveň $2 \equiv S \equiv 4 \pmod{n}$, čo nie je možné, lebo $2 \equiv 4 \pmod{n}$ len pre $n \leq 2$. Opäť dostávame spor.

Tvrdenie je teda dokázané, hľadaný rozklad neexistuje.

PIATA SÉRIA

5.1 (*Peter Kozák*) Predpokladajme, že x je koreňom danej rovnice, teda pre x platí:

$$x^4 + ax^3 + 2x^2 + bx + 1 = 0.$$

Očividne $x \neq 0$, a teda platí

$$ax^2 + b = -\frac{x^4 + 2x^2 + 1}{x} = -\frac{(x^2 + 1)^2}{x}.$$

Potom z *Cauchyho nerovnosti*

$$|\vec{u}|^2 \cdot |\vec{v}|^2 \geq \vec{u} \cdot \vec{v} \quad \text{alebo} \quad (u_1^2 + u_2^2) \cdot (v_1^2 + v_2^2) \geq (u_1v_1 + u_2v_2)^2.$$

pre vektory $(u_1, u_2) = \vec{u} = (a, b)$ a $(v_1, v_2) = \vec{v} = (x^2, 1)$ dostávame:

$$(a^2 + b^2) \cdot (x^4 + 1) \geq (ax^2 + b)^2 = \frac{(x^2 + 1)^4}{x^2}.$$

Z toho (výraz $x^4 + 1$ je zrejme vždy kladný)

$$a^2 + b^2 \geq \frac{(x^2 + 1)^4}{x^2(x^4 + 1)}. \quad (1)$$

Teraz dokážeme, že platí

$$\frac{(x^2 + 1)^4}{x^2(x^4 + 1)} \geq 8,$$

pre každé reálne číslo x rôzne od 0. Ekvivalentnými úpravami tejto nerovnosti postupne dostávame

$$\begin{aligned} (x^2 + 1)^4 &\geq 8x^2(x^4 + 1), \\ x^8 - 4x^6 + 6x^4 - 4x^2 + 1 &\geq 0, \\ (x^2 - 1)^4 &\geq 0, \end{aligned}$$

čo zrejme platí pre každé $x \in \mathbb{R}$. Spolu s (1) potom máme $a^2 + b^2 \geq 8$. Na druhej strane, rovnica $x^4 + 2x^3 + 2x^2 + 2x + 1 = 0$ má reálny koreň $x = -1$, a pre ňu $a = b = 2$, čiže $a^2 + b^2 = 8$. To znamená, že hľadaná minimálna hodnota súčtu $a^2 + b^2$ je 8.

5.2 (*Viera Růžičková*) Nech $n^2 = a^3 - (a-1)^3$ pre nejaké $a \in \mathbb{N}$, $a > 1$. Jednoduchou úpravou dostávame $3a^2 - 3a + 1 - n^2 = 0$. Z toho potom $a = \frac{3 + \sqrt{12n^2 - 3}}{6}$. Pretože a

je prirodzené číslo, musí byť $12n^2 - 3$ štvorec. Platí $12n^2 - 3 = 3(2n-1)(2n+1)$. Najväčší spoločný deliteľ čísel $2n-1$ a $2n+1$ je 1, pretože sú obe nepárne a ich rozdiel je 2. Ak je teda číslo $3(2n-1)(2n+1)$ druhou mocninou prirodzeného čísla, potom jedno z čísel $2n-1$ a $2n+1$ je tiež druhou mocninou a druhé z týchto čísel je rovné trojnásobku druhej mocniny prirodzeného čísla. Pretože sú obe čísla $2n \pm 1$ nepárne, môžu nastať dva prípady:

1) Platí $2n-1 = (2m-1)^2$ pre nejaké prirodzené číslo m . Potom

$$2n-1 = 4m^2 - 4m + 1, \quad \text{čiže} \quad n = 2m^2 - 2m + 1 = m^2 + (m-1)^2.$$

2) Platí $2n-1 = 3 \cdot (2m-1)^2$ pre nejaké prirodzené číslo m . Potom $2n+1 = 3 \cdot (2m-1)^2 + 2$, a zároveň má byť $2n+1$ štvorcom. Lenže žiaden štvorec prirodzeného čísla nedáva po delení tromi zvyšok 2, a teda tento prípad nenastáva.

Tým sme dokázali, že vyhovujúce n je vždy súčtom dvoch štvorcov po sebe idúcich prirodzených čísel. Príklady takých n sú napr.

$$13^2 = 8^3 - 7^3, \quad 13 = 2^2 + 3^2; \quad 181^2 = 105^3 - 104^3, \quad 181 = 9^2 + 10^2.$$

5.3 (*Róbert Macho*) Uvedomme si, že funkcia f nadobúda len celočíselné hodnoty. Zaoberajme sa najprv minimom. Zrejme pre každé $n \in \mathbb{N}$ platí:

$$f(n) = n - \sum_{k=1}^m [r_k n] \geq n - \sum_{k=1}^m r_k n = 0.$$

To znamená, že minimum funkcie f bude nezáporné. Ďalej ukážeme, že je rovné 0 (pre ľubovoľné r_1, r_2, \dots, r_m spĺňajúce podmienky zadania). Položme $r_k = \frac{p_k}{q_k}$, kde

$p_k, q_k \in \mathbb{N}$, $(p_k, q_k) = 1$, pre $k = 1, 2, \dots, m$. Ďalej označme $N = \text{nsn}(q_1, q_2, \dots, q_m)$ (najmenší spoločný násobok) a nech $N = q_k s_k$, $s_k \in \mathbb{N}$, pre $k = 1, 2, \dots, m$. Potom zrejme $r_k = \frac{p_k s_k}{N}$, pričom $\sum_{k=1}^m p_k s_k = N$. Ďalej platí:

$$f(N) = N - \sum_{k=1}^m \left[\frac{p_k s_k}{N} \cdot N \right] = N - \sum_{k=1}^m [p_k s_k] = N - \sum_{k=1}^m p_k s_k = N - N = 0.$$

Tým sme ukázali, že minimum funkcie f je rovné 0 pre každé r_1, r_2, \dots, r_m spĺňajúce podmienky zadania.

Zaoberajme sa teraz maximom. Keďže $\forall x \in \mathbb{R} : [x] > x - 1$, tak

$$\forall n \in \mathbb{N} : f(n) = n - \sum_{k=1}^m [r_k n] < n - \sum_{k=1}^m (r_k n - 1) = n - n \cdot \sum_{k=1}^m r_k + m = m.$$

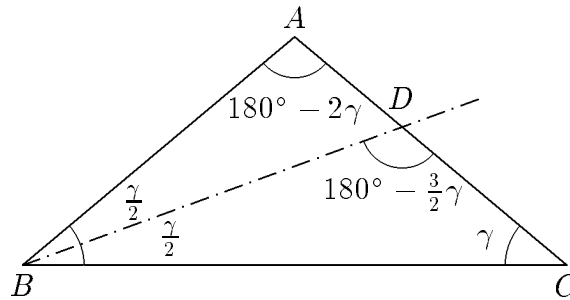
Preto $\forall n \in \mathbb{N} : f(n) \leq m - 1$. Ukážeme, že pre ľubovoľné r_1, r_2, \dots, r_m existuje $n \in \mathbb{N}$ také, že $f(n) = m - 1$, a teda $m - 1$ je maximum funkcie f . Stačí zvoliť $n = N - 1$. Potom

$$\begin{aligned} f(N - 1) &= N - 1 - \sum_{k=1}^m \left\lfloor \frac{p_k s_k}{N} \cdot (N - 1) \right\rfloor = N - 1 - \sum_{k=1}^m \lfloor p_k s_k - r_k \rfloor = \\ &= N - 1 - \sum_{k=1}^m (p_k s_k - 1) = N - 1 - N + m = m - 1. \end{aligned}$$

Teda maximum funkcie f je rovné $m - 1$ pre ľubovoľné r_1, r_2, \dots, r_m spĺňajúce podmienky zadania.

5.4 Označme $\gamma = |\sphericalangle ABC| = |\sphericalangle ACB|$. Zrejme $\gamma \in (0^\circ, 90^\circ)$. Potom pre veľkosti ostatných uhlov platí: $|\sphericalangle CBD| = |\sphericalangle DBA| = \frac{1}{2}\gamma$, $|\sphericalangle BAC| = 180^\circ - 2\gamma$, $|\sphericalangle BDC| = 180^\circ - \frac{3}{2}\gamma$ (obr. 26). Zo sínusovej vety pre trojuholníky ABD a BDC dostávame:

$$|AD| = |BD| \cdot \frac{\sin \frac{1}{2}\gamma}{\sin 2\gamma} \quad \text{a} \quad |BC| = |BD| \cdot \frac{\sin \frac{3}{2}\gamma}{\sin \gamma}.$$



Obr. 26

Dosadíme tieto vzťahy do $|BC| = |BD| + |AD|$ a úpravami dostávame:

$$\begin{aligned} |BD| \cdot \frac{\sin \frac{3}{2}\gamma}{\sin \gamma} &= |BD| + |BD| \cdot \frac{\sin \frac{1}{2}\gamma}{\sin 2\gamma}, \\ 2 \cdot \sin \frac{3}{2}\gamma \cdot \cos \gamma &= \sin 2\gamma + \sin \frac{1}{2}\gamma, \\ \sin \frac{5}{2}\gamma - \sin 2\gamma &= 0, \quad \text{a teda} \quad 2 \cdot \sin \frac{1}{4}\gamma \cdot \cos \frac{9}{4}\gamma = 0. \end{aligned}$$

Použili sme vzorce pre súčet a rozdiel dvoch sínusov. Riešeniami poslednej rovnice sú

$$\gamma = 4k \cdot 180^\circ = k \cdot 720^\circ \quad \text{a} \quad \gamma = \frac{4}{9}(90^\circ + l \cdot 180^\circ) = 40^\circ + l \cdot 80^\circ,$$

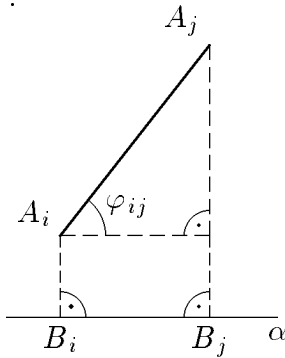
kde $k, l \in \mathbb{Z}$. Keďže $\gamma \in (0^\circ, 90^\circ)$, jediným riešením je $\gamma = 40^\circ$, a teda $|\sphericalangle BAC| = 180^\circ - 2\gamma = 100^\circ$. Skúškou sa ľahko presvedčíme o správnosti výsledku.

5.5 Označme $A_1 A_2 A_3 A_4$ daný pravidelný štvorsten. Nech $|A_1 A_2| = 1$. Uložme ho tak, aby ležal nad vodorovnou rovinou α (v jednom pevne zvolenom polpriestore určenom touto rovinou). Nech B_1, B_2, B_3 a B_4 sú po rade kolmé priemety bodov A_1, A_2, A_3 a A_4 do roviny α . Označme $v_i = |A_i B_i|$ pre $i = 1, 2, 3, 4$. Bez ujmy na všeobecnosti nech $v_1 \leq v_2 \leq v_3 \leq v_4$, a nech φ_{ij} , $i \neq j$ je veľkosť uhla, ktorý zvierajú hrana $A_i A_j$ s rovinou α (obr. 27). Potom

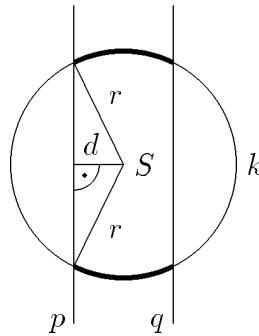
$$\sin \varphi_{ij} = \frac{|v_i - v_j|}{|A_i A_j|} = |v_i - v_j|.$$

Zrejme $\varphi_{ij} \in \langle 0^\circ, 90^\circ \rangle$ (pre $1 \leq i < j \leq 4$). Pretože funkcia sínus je na tomto intervale rastúca, tak φ_{ij} je maximálne práve vtedy, keď je maximálne $|v_i - v_j|$, a to je maximálne pre $i = 1, j = 4$. Teda $\varphi = \varphi_{14}$. Pre jednoduchosť zvolíme $v_1 = 0$, čiže $A_1 = B_1 \in \alpha$.

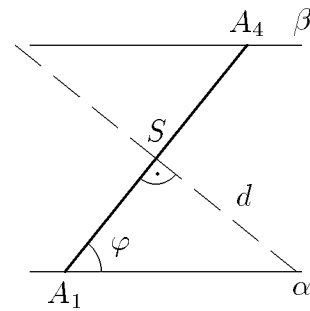
Ukážeme, že najmenšia možná hodnota $\varphi_{min} = 45^\circ$. Táto hodnota sa nadobúda v prípade, keď $A_2 \in \alpha$, $\overleftrightarrow{A_3 A_4} \parallel \alpha$. Vtedy je $|v_4 - v_1|$ rovné vzdialenosti priamok $\overleftrightarrow{A_1 A_2}$ a $\overleftrightarrow{A_3 A_4}$. Lahko možno určiť, že táto vzdialenosť je $\frac{\sqrt{2}}{2}$, a teda $\sin \varphi = \frac{\sqrt{2}}{2}$, čo dáva $\varphi = 45^\circ$.



Obr. 27



Obr. 28



Obr. 29

Teraz sporom ukážeme, že $\varphi \geq 45^\circ$. Uvažujme polohu štvorstena $A_1 A_2 A_3 A_4$ (pri ktorej $0 = v_1 \leq v_2 \leq v_3 \leq v_4$) takú, že $\varphi < 45^\circ$, čiže $v_4 - v_1 < \frac{\sqrt{2}}{2}$. Vezmime si teraz rovinu β rovnobežnú s α takú, že $A_4 \in \beta$ a označme S stred $A_1 A_4$. Ďalej vezmime rovinu γ kolmú na $\overleftrightarrow{A_1 A_4}$ takú, že $S \in \gamma$. Označme p, q prieniky roviny γ s rovinami α, β . Potom zrejme $A_2, A_3 \in \gamma$ (pretože $|A_1 A_3| = |A_4 A_3|$, $|A_1 A_2| = |A_4 A_2|$) a γ je množina bodov s rovnakou vzdialenosťou od bodov A_1, A_4 , $|S A_2| = |S A_3| = \frac{\sqrt{3}}{2}$ (dĺžka ťažnice v rovnostrannom trojuholníku), a teda body A_2, A_3 ležia na kružnici k ($S, r = \frac{\sqrt{3}}{2}$) v rovine γ . Naviac body A_2, A_3 ležia medzi rovinami α, β , a teda v rovine γ ležia medzi priamkami p, q (situácia v rovine γ je na obr. 28). Vzdialenosť $d = |S p| = |S q|$ môžeme určiť z obr. 29 :

$$\operatorname{tg} \varphi = \frac{d}{|AS|} = \frac{d}{\frac{1}{2}} = 2d, \quad \text{teda} \quad d = \frac{1}{2} \operatorname{tg} \varphi < \frac{1}{2} \operatorname{tg} 45^\circ = \frac{1}{2}.$$

Využili sme rastúcosť funkcie $\operatorname{tg} x$ na intervale $\langle 0^\circ, 45^\circ \rangle$. Potom body A_2, A_3 ležia na zjednotení kružnicových oblúkov vyznačených na obr. 28 (ďalej len oblúkov). Zrejme nemôžu ležať na tom istom oblúku, lebo potom by bolo $|A_2 A_3| \leq 2d < 1$, čo by bol

spor. Teda ležia na rôznych oblúkoch a potom (tetiva je tým kratšia, čím menší uhol ju určuje):

$$|A_2 A_3| \geq 2\sqrt{r^2 - d^2} = 2\sqrt{\frac{3}{4} - d^2} > 2\sqrt{\frac{3}{4} - \frac{1}{4}} = 2\sqrt{\frac{1}{2}} = \sqrt{2} > 1,$$

čo je spor. Tým sme ukázali, že najmenšia možná hodnota φ je $\varphi_{min} = 45^\circ$.

5.6 (*Miroslav Dudík*) Tvrdenie budeme dokazovať sporom. Nech je postupnosť $\{a_n\}$ periodická s periódou p (na začiatok zvolíme p ako najmenšiu periódu).

Potom $a_n = a_{n+p}$, pre každé prirodzené číslo n , teda

$$\lfloor x^{n+1} \rfloor - x \cdot \lfloor x^n \rfloor = \lfloor x^{n+p+1} \rfloor - x \cdot \lfloor x^{n+p} \rfloor,$$

čo po jednoduchej úprave dáva

$$\lfloor x^{n+p+1} \rfloor - \lfloor x^{n+1} \rfloor = x \cdot (\lfloor x^{n+p} \rfloor - \lfloor x^n \rfloor). \quad (1)$$

Výraz $\lfloor x^{n+s} \rfloor - \lfloor x^n \rfloor$ je zrejme pre dostatočne veľké s kladný pre všetky $n \in \mathbb{N}$. Keďže vzťah (1) platí aj pre každý násobok periódy p , môžeme (prípadným prechodom k jej násobkom) bez ujmy na všeobecnosti predpokladať, že $\lfloor x^{n+p} \rfloor - \lfloor x^n \rfloor > 0$ pre každé $n \in \mathbb{N}$. Potom

$$x = \frac{\lfloor x^{n+p+1} \rfloor - \lfloor x^{n+1} \rfloor}{\lfloor x^{n+p} \rfloor - \lfloor x^n \rfloor} \in \mathbb{Q},$$

čiže x je racionálne číslo, ktoré možno zapísať v základnom tvare $x = \frac{r}{q}$, kde $r, q \in \mathbb{N}$, $(r, q) = 1$. Ak potom $\lfloor x^{n+p} \rfloor - \lfloor x^n \rfloor = A \cdot q^a$, $q \nmid A$, $a \in \mathbb{N}$, tak zrejme $q \neq 1$, pretože $x \notin \mathbb{Z}$.

Položme teraz vo vzťahu (1) $n = m + 1$. Dostávame

$$\lfloor x^{m+p+2} \rfloor - \lfloor x^{m+2} \rfloor = x \cdot (\lfloor x^{m+p+1} \rfloor - \lfloor x^{m+1} \rfloor) = x^2 \cdot (\lfloor x^{m+p} \rfloor - \lfloor x^m \rfloor).$$

Poslednú rovnosť sme dostali z (1) položením $n = m$. Matematickou indukciou možno teraz jednoducho dokázať, že

$$\lfloor x^{m+p+k} \rfloor - \lfloor x^{m+k} \rfloor = x^k \cdot (\lfloor x^{m+p} \rfloor - \lfloor x^m \rfloor) = x^k \cdot A \cdot q^a,$$

kde k je ľubovoľné prirodzené číslo. Pre $k = a + 1$ máme

$$\lfloor x^{m+p+a+1} \rfloor - \lfloor x^{m+a+1} \rfloor = x^{a+1} \cdot A \cdot q^a = \frac{p^{a+1}}{q^{a+1}} \cdot A \cdot q^a = \frac{A \cdot p^{a+1}}{q}.$$

Rozdiel celých čísel $\lfloor x^{m+p+a+1} \rfloor$ a $\lfloor x^{m+a+1} \rfloor$ musí však byť celý, teda $q \mid A \cdot p^{a+1}$, čo však nie je možné, a dostávame spor. Postupnosť $\{a_n\}$ teda nemôže byť periodická.

5.7 (*Viera Růžičková, Pavol Novotný, Keszegh Balázs*)

Zo zadaného vzťahu $|AX| \cdot |BN| = 2 \cdot |BX| \cdot |CN|$ dostávame

$$\frac{|AX|}{|BX|} = 2 \cdot \frac{|CN|}{|BN|}, \quad \text{čiže} \quad \frac{|AX| + |BX|}{|BX|} = \frac{2 \cdot |CN| + |BN|}{|BN|},$$

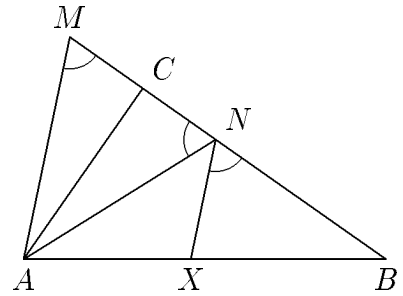
teda

$$\frac{|AB|}{|BX|} = \frac{2 \cdot |CN| + |BN|}{|BN|}.$$

Označme teraz M bod na polpriamke \overrightarrow{BC} , pre ktorý platí $|BM| = 2|CN| + |BN|$. Potom platí

$$\frac{|AB|}{|BX|} = \frac{|BM|}{|BN|}.$$

Preto sú úsečky AM a XN rovnolehlé zo stredom rovnolehlosti v B a teda aj rovnobežné. Pre uhly potom platí $|\sphericalangle AMB| = |\sphericalangle XNB|$. Zo zadania ďalej $|\sphericalangle BNX| = |\sphericalangle ANC|$, teda $|\sphericalangle AMB| = |\sphericalangle ANC|$. Trojuholník ANM je preto rovnoramenný a platí $|AM| = |AN|$. Keďže $|BM| = 2|CN| + |BN|$, zrejme aj $|CM| = |CN|$. Potom je ale AC výškou a zároveň ťažnicou v rovnoramennom trojuholníku ANM , teda $|\sphericalangle ACB| = 90^\circ$.



Obr. 30

Obsah

DRUHÝ DIEL

Kategória P	1
Zadania súťažných úloh	1
Riešenia súťažných úloh	13
Štvrtá Stredoeurópska olympiáda v informatike	29
Zadania súťažných úloh	30
9. Medzinárodná olympiáda v informatike	35
Zadania súťažných úloh	36
Korešpondenčný seminár SK MO	46
Zadania súťažných úloh	47
Riešenia súťažných úloh	52
Obsah	85

RNDr. Karel Horák, CSc. – Richard Kollár
Jana Višňovská – Tomáš Vinař
Bronislava Brejová – Úlohová komisia MO

**Štyridsiatyšiesty ročník
matematickej olympiády
na stredných školách
2. diel**

Vydala IUVENTA v roku 1998
Sadzbu programom $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{T}\mathcal{E}\mathcal{X}$ pripravili RNDr. Karel Horák, CSc. a Richard Kollár
Grafická úprava obálky Karel Horák a Richard Kollár
Neprešlo jazykovou úpravou
1. vydanie

ISBN 80-88893-19-4