



Informácie a pravidlá

Pre koho je súťaž určená?

- Do **kategórie B** sa smú zapojiť len tí žiaci základných a stredných škôl, ktorí ešte ani v tomto, ani v nasledujúcom školskom roku nebudú končiť strednú školu.
- Do **kategórie A** sa môžu zapojiť všetci žiaci (základných aj) stredných škôl.

Odvzdávanie riešení domáceho kola

Riešitelia domáceho kola odovzdávajú riešenia sami, v elektronickej podobe, a to priamo na stránke olympiády: <http://oi.sk/>. Odovzdávanie riešení bude spustené najneskôr do konca septembra.

Termíny 41. ročníka OI

- Riešenia domáceho kola je potrebné v kategórii A odovzdať najneskôr **15. 11. 2025**, v kategórii B najneskôr **30. 11. 2025**.
- Krajské kolo oboch kategórií prebehne 20. 1. 2026 zvlášť v každom kraji.
- Celoštátne kolo kategórie A sa bude konať v dňoch 18.-21. 3. 2026 (súťaží sa vo štvrtok a piatok).
- Detailnejšie informácie o priebehu súťaže nájdete na webe OI: <https://oi.sk/?s=pravidla>

Ako majú vyzerat riešenia úloh?

V **praktických úlohách** je vašou úlohou vytvoriť program, ktorý bude riešiť zadanú úlohu. Program musí byť v prvom rade korektný a funkčný, v druhom rade sa snažte aby bol čo najefektívnejší.

V kategórii B môžete použiť ľubovoľný programovací jazyk.

V kategórii A musíte riešenia praktických úloh písať v jednom z podporovaných jazykov (napr. C++, Python alebo Java). Presný zoznam podporovaných jazykov vrátane konkrétnych verzií ich kompilátorov a interpreterov nájdete v systéme na odovzdávanie riešení. Taktiež tam nájdete presnejší popis toho, ako sa majú vaše programy správať, napr. detaily realizácie vstupu a výstupu.

Odovzdaný program bude automaticky otestovaný na viacerých vopred pripravených testovacích vstupoch. Podľa toho, na koľko z nich dá správnu odpoveď, vám budú pridelené body. Výsledok testovania sa dozviete krátko po odovzdaní. Ak váš program nezíska plný počet bodov, budete ho môcť vylepšiť a odovzdať znova, až do uplynutia termínu na odovzdávanie.

Ak nie je v zadaní povedané ináč, riešenia **teoretických úloh** musia obsahovať:

- podstatné časti algoritmu ako program (v ľubovoľnom bežnom prog. jazyku) alebo ekvivalentný pseudokód
- podrobný slovný popis použitého algoritmu
- zdôvodnenie jeho správnosti
- diskusiu o efektivite zvoleného riešenia (odhad časovej a pamätovej zložitosti)

Ak používate v programe netriviálne algoritmy alebo dátové štruktúry (napr. rôzne súčasti STL v C++), súčasťou slovného popisu algoritmu by mal byť stručný popis ich implementácie.

Usporiadateľ súťaže

Olympiádu v informatike (OI) vyhlasuje *Ministerstvo školstva SR* v spolupráci so *Slovenskou informatickou spoločnosťou* (odborným garantom súťaže) a *Slovenskou komisiou Olympiády v informatike*. Súťaž organizuje *Slovenská komisia OI* a v jednotlivých krajoch ju riadia *krajské komisie OI*. Na jednotlivých školách ju zaisťujú učitelia informatiky. Celoštátne kolo OI, tlač materiálov a ich distribúciu po organizačnej stránke zabezpečuje NIVAM v tesnej súčinnosti so Slovenskou komisiou OI.



B-I-1 Preteky

Toto je praktická úloha. Riešiť sa dá v ľubovoľnom programovacom jazyku, odovzdávajú sa len správne riešenia pre vopred pripravené testovacie dáta. Detailnejšie pokyny k odovzdávaniu riešení sú uvedené nižšie.

Ich revírom je pole za dedinou. Ich tempo je prinajlepšom svižné. To sú chlapci na traktoroch z Kocúrkova.

V Kocúrkove tento rok organizuje Klub Sebestačných Poľnohospodárov prestížnu súťaž o najrýchlejší traktor. V Kocúrkove sa však nachádza len jediná cesta, na konci ktorej je rýchlostný radar. Pretekári si preto tento rok budú musieť dávať pozor na svoju rýchlosť pri prechode cez cieľovú čiaru.

Súťažná úloha

Trať pretekov si predstavíme ako úsečku dĺžky n metrov. Traktor si reprezentujeme hmotným bodom, ktorý stojí na začiatku trate a má počiatočnú rýchlosť $v = 0$ m/s. Na začiatku každej sekundy môže vodič traktora pridať, čím zvýši svoju rýchlosť v o 1 m/s; pribrzdiť, čím svoju rýchlosť v zníži o 1 m/s; alebo ostať na rovnakej rýchlosti v . Následne ide traktor rýchlosťou v jednu sekundu dopredu po trati, čím sa posunie o v metrov. Zmeny rýchlosti sa dejú okamžite.

Preteky končia na konci tej sekundy, počas ktorej traktor prešiel cieľovú čiaru (koniec úsečky). Postačuje aj, ak sa traktor na konci tejto sekundy nachádza priamo na cieľovej čiare.

Na cieľovej čiare je namontovaný rýchlostný radar. V okamihu, keď sa traktor nachádza na cieľovej čiare, nemôže mať rýchlosť väčšiu ako X m/s.

Organizátori vedia požiadať políciu o nastavenie maximálnej povolenej rýchlosti v celi, ktorú meria radar. Zaujímajú ich preto, za aký najmenší čas sa dajú preteky dokončiť pre zadanú maximálnu rýchlosť v v celi X . A aby vedeli zvoliť divácky najzaujímavejšiu hodnotu, chcú by to vedieť pre q rôznych maximálnych rýchlostí v v celi.

Formát vstupu a výstupu

V prvom riadku vstupu sú čísla n a q : dĺžka trate a počet otázok organizátorov. Zvyšok vstupu tvorí q riadkov. V i -tom z nich je číslo X_i : maximálna rýchlosť, ktorou môže traktor prejsť cez cieľ.

Pre každú otázku vypíšte na jeden riadok najmenší počet sekúnd, za ktorý vie traktor dokončiť preteky.

Obmedzenia a hodnotenie

Táto úloha má 5 hodnotených vstupov. Za každý správne vyriešený vstup dostanete dva body.

Vo všetkých vstupoch platí, že $1 \leq n \leq 10^8$, $1 \leq q \leq 1000$, a $1 \leq X_i \leq 10^8$.

Niektoré vstupy sú menšie a ľahšie riešiteľné ako iné. Presnejšie:

- V prvom vstupe má trať len $n = 15$ metrov a pýtame sa $q = 5$ otázok.
- V druhom vstupe má trať $n = 200$ metrov, pýtame sa $q = 5$ otázok a v každej je $X_i \geq 20$.
- V treťom vstupe platí $n = 300$ a $q = 20$.

Odovzdávanie riešení

Zo stránky <https://oi.sk/?d=rocnik> si stiahnite ZIP archív obsahujúci 5 testovacích vstupov.

Vstupy sú nazvané 01.txt až 05.txt.

Vyrobte k čo najviac vstupom správne výstupy a uložte ich do súborov sol01.txt až sol15.txt.

Odovzdajte ZIP archív obsahujúci zdrojový kód vášho programu a tieto výstupné súbory.

Za každý správny výstupný súbor získate 2 body.



Príklad

vstup

```
10 3
1
3
4
```

výstup

```
6
5
4
```

Pre $X = 1$ je najlepšie ísť napr. nasledovne:

- sekunda 1: zrýchliť na 1 m/s, prejsť na meter 1
- sekunda 2: zrýchliť na 2 m/s, prejsť na meter 3
- sekunda 3: zrýchliť na 3 m/s, prejsť na meter 6
- sekunda 4: spomaliť na 2 m/s, prejsť na meter 8
- sekunda 5: spomaliť na 1 m/s, prejsť na meter 9
- sekunda 6: zostať na 1 m/s, prejsť na meter 10

Pre $X = 3$ je najlepšie ísť napr. nasledovne:

- sekunda 1: zrýchliť na 1 m/s, prejsť na meter 1
- sekunda 2: zrýchliť na 2 m/s, prejsť na meter 3
- sekunda 3: zrýchliť na 3 m/s, prejsť na meter 6
- sekunda 4: zostať na 3 m/s, prejsť na meter 9
- sekunda 5: zostať na 3 m/s, prejsť na meter 12



B-I-2 Nedočkavosť

Toto je praktická úloha. Riešiť sa dá v ľubovoľnom programovacom jazyku, odovzdávajú sa len správne riešenia pre vopred pripravené testovacie dáta. Detailnejšie pokyny k odovzdávaniu riešení sú uvedené nižšie.

Začal sa školský rok a žiaci opäť zasadli do školských lavíc. Niektorí žiaci riešia olympiádu v informatike a netrpezlivo čakajú na zverejnenie zadanií. Tí najnetrpzelivejší dokonca neustále obnovujú webovú stránku. V momente, ako sa zadania objavia na stránke, sa to títo žiaci hneď dozvedia. Následne sa informácia o zverejnených zadaniach začne šíriť po triede.

Vašou úlohou bude zistiť, ako rýchlo sa táto informácia rozšírila.

Súťažná úloha

Triedu si môžeme predstaviť ako mriežku $r \times s$. V každom políčku mriežky (lavici) sedí jeden žiak jedného z troch typov:

- Žiak K – olympiádu **rieši** a netrpezlivo **kontroluje** stránku
- Žiak n – olympiádu **rieši**, stránku však vôbec **nekontroluje**
- Žiak # – olympiádu **nerieši**

V minúte 0 boli zverejnené zadania súťaže, všetci žiaci K sa o tom okamžite dozvedeli a začali túto informáciu šíriť.

Šírenie informácie o zadaniach prebieha nasledovne: Na začiatku každej minúty (počnúc minútou 1) povie každý žiak, ktorý olympiádu **rieši** (teda K a n) a **vie**, že zadania sú zverejnené, túto informáciu všetkým svojim susedom. Dvaja žiaci sú susedia, ak ich lavice v mriežke zdieľajú hranu. Každý žiak má teda najviac štyroch susedov – naľavo, napravo, hore a dole. Žiaci na krajoch mriežky ich majú menej ako štyroch.

Žiaci, ktorí olympiádu *neriešia* sa o jej zadaniach síce dozvedia, ďalej túto informáciu však *nešíria*.

Pri takomto šírení sa postupne čoraz viac a viac žiakov v triede dozvedá informáciu o zadaniach. Môžete si však rozmyslieť, že po dostatočne dlhom čase sa už nikto nový o zadaniach nedozvie; tí ktorí o nich ešte nevedia sa o nich už nikdy nedozvedia (aspoň nie týmto spôsobom).

O každom žiakovi, ktorý sa o zadaniach dozvedel, si vieme zistiť minútu, v ktorej sa to stalo prvýkrát. Napríklad všetci žiaci označení ako K sa to dozvedeli v minúte 0.

Pre ľubovoľnú minútu x teda vieme spočítať: *počet žiakov, ktorí sa o zadaniach olympiády dozvedeli prvýkrát v minúte x* . Tento počet si označíme p_x . A práve tieto hodnoty nás zaujímajú v tejto úlohe.

Formát vstupu a výstupu

Na vstupe dostanete najprv dve celé čísla $1 \leq r, s \leq 1000$ – počet riadkov a stĺpcov triedy.

Nasleduje r riadkov, z ktorých každý obsahuje s znakov K, n alebo #, ktoré opisujú sedenie žiakov v triede.

Nech m je *posledná* minúta, v ktorej sa o zadaniach olympiády dozvedel nejaký nový žiak. Je to teda najväčšia hodnota, pre ktorú je p_m nenulové. Na prvý riadok výstupu vypíšte hodnotu m .

Následne vypíšte $m + 1$ riadkov. Na i -ty z nich, číslovaný od 0, vypíšte hodnotu p_i . Vypísať teda máte počty žiakov, ktorí sa o zadaniach dozvedeli prvýkrát v minúte i , pre všetky i od 0 po m (vrátane).

Obmedzenia a hodnotenie

Táto úloha má 10 hodnotených vstupov. Za každý správne vyriešený vstup dostanete jeden bod. Pre jednotlivé vstupy platia nasledovné dodatočné obmedzenia:

- Vstup 1: Mriežka má rozmer 5×6 .
- Vstup 2: Mriežka má rozmer 1×1000 .
- Vstupy 3 a 4: V mriežke je práve 1 žiak typu K.



- Vstupy 5 a 6: Mriežka má rozmer 100×100 .
- Vstup 7: Umiestnenie žiakov typu # v mriežke vytvára špecifický “hadovitý” vzor. (Odporúčame si pozrieť bližšie konkrétny vstup.)
- Vstup 8: V mriežke je práve 1000 žiakov typu K.
- Vstup 9: V mriežke nie je žiaden žiak typu #.
- Vstup 10: Bez dodatočných obmedzení.

Príklad

vstup

```
3 5
Knnn#
#n#Kn
n#nnK
```

výstup

```
3
3
6
4
1
```

V čase 0 sa o zadaniach dozvedia traja žiaci, ktorí sedia na pozíciách označených K.

V čase 1 sa o zverejnení dozvedia ich susedia. Všimnime si, že so zadaniami otravujú aj dvoch žiakov, ktorí olympiádu neriešia. Tiež si všimnime, že dvaja žiaci sa o zadania dozvedia od dvoch susedov naraz.

V čase 3 sa ako posledný o zadaniach dozvie žiak, ktorý nerieši olympiádu v spodnom riadku.

Nakoniec si všimnime, že žiak v ľavom dolnom rohu sa o zverejnení nikdy nedozvie.

Odvzdávanie riešení

Zo stránky <https://oi.sk/?d=rocnik> si stiahnete ZIP archív obsahujúci 10 testovacích vstupov.

Vstupy sú nazvané 01.txt až 10.txt.

Vyrobte k čo najviac vstupom správne výstupy a uložte ich do súborov sol01.txt až sol10.txt.

Odvzdajte ZIP archív obsahujúci **zdrojový kód vášho programu** a tieto výstupné súbory.

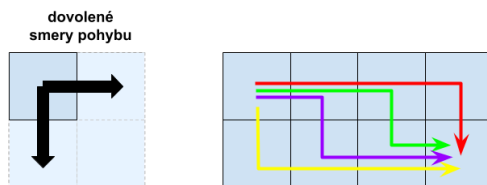
Za každý správny výstupný súbor získate 1 body.



B-I-3 Cesty v mriežke

Toto je **teoretická úloha**. Pomocou webového rozhrania odovzdaj **súbor vo formáte PDF**, obsahujúci riešenie, spĺňajúce požiadavky uvedené v pravidlách.

Na konci prázdnin sa Hanka už doma poriadne nudila. Aby ju jej mama zabavila, nakreslila jej štvorčekovú mriežku s r riadkami a s stĺpcami a spýtala sa jej: “Koľko rôznych ciest vedie z *ľavého horného rohu* do *pravého dolného rohu*, ak sa môžeš hýbať iba doprava alebo dodola?” Dúfala, že počítanie všetkých ciest bude Hanke trvať dlho.



Obr. 1: Na obrázku sú znázornené všetky 4 rôzne cesty v mriežke veľkosti 2×3 .

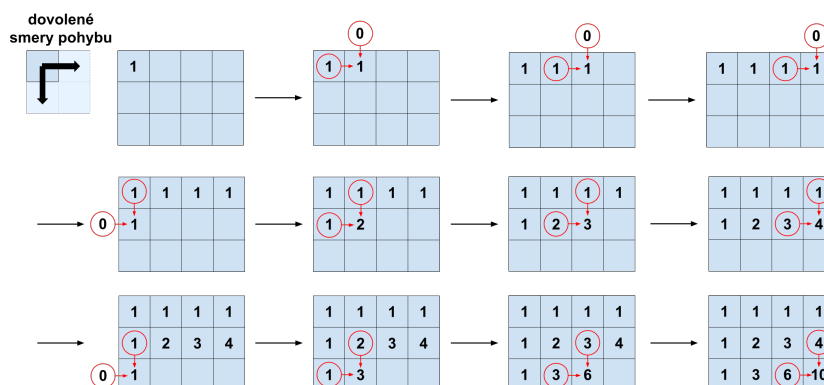
Hanka je však dušou informatička a vymyslela oveľa šikovnejšie riešenie, ako cesty počítať. Uvedomila si totiž, že nemusí počítať iba výslednú hodnotu, pomôcť jej vedú aj priebežné výpočty, ktoré si zapíše do políček mriežky.

Na políčko v riadku x a stĺpci y (označíme ho ako (x, y)) si chce zapísať počet rôznych ciest, ktoré začínajú v ľavom hornom rohu a končia práve na políčku (x, y) . Na začiatku vie, že pre začiatkové políčko $(1, 1)$ je táto hodnota 1, keďže vie iba stáť na mieste.

Všimla si, že ďalšie hodnoty vie jednoducho spočítať. Na políčko (x, y) sa totiž dá dostať iba dvoma spôsobmi – buď príde na políčko $(x - 1, y)$ a posunie sa dodola, alebo príde na políčko $(x, y - 1)$ a posunie sa doprava. To ale znamená, že počet rôznych ciest na políčko (x, y) sa dá spočítať ako súčet ciest na políčko $(x - 1, y)$ a $(x, y - 1)$, teda súčet čísel z políčka nad ním a naľavo od neho.

Jediné čo potrebuje je, aby pri počítaní hodnoty na políčku (x, y) mala už zapísané čísla na oboch políčkach $(x - 1, y)$ a $(x, y - 1)$. Na to však stačí, aby čísla do mriežky dopĺňala postupne riadok po riadku, každý riadok zľava doprava.

Výsledná hodnota, na ktorú sa pýta mama, bude na konci zapísaná v pravom dolnom políčku.



Obr. 2: Postup znázorňujúci Hankino počítanie počtu ciest na mriežke 3×4 .

Takýto postup sa dokonca dá jednoducho zapísať aj ako program, takže Hanka vie zrazu riešiť ľubovoľne veľké mriežky veľmi rýchlo.



Listing programu (Python)

```
r, s = 3, 4
# prázdne pole s r riadkami a s stĺpcami
mriezka = [[0 for _ in range(s)] for _ in range(r)]

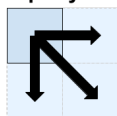
mriezka[0][0] = 1 # počiatočná hodnota

for x in range(r):
    for y in range(s):
        if x != 0: # existuje políčko nadomnou
            mriezka[x][y] += mriezka[x - 1][y]
        if y != 0: # existuje políčko naľavo
            mriezka[x][y] += mriezka[x][y - 1]

print(mriezka[r - 1][s - 1])
```

Aby sa opäť nezačala nudiť, začala Hankina mama vymýšľať rôzne alternatívy tejto úlohy. Napríklad, čo ak by bolo dovolené sa v jednom kroku posunúť šikmo doprava-dole? Hankine algoritmické riešenie je však prekvapivo veľmi flexibilné a vyriešiť takýto variant je veľmi jednoduché. Stačí si uvedomiť, že iba pribudla dodatočná možnosť ako sa dostať na políčko (x, y) . Hodnotu, ktorú zapíšeme na políčko (x, y) , opäť určíme ako súčet hodnôt na políčkach, z ktorých sme mohli spraviť posledný krok: $(x - 1, y)$, $(x, y - 1)$ a navyše aj $(x - 1, y - 1)$.

dovolené smery
pohybu



1	1	1	1
1	3	5	7
1	5	13	

Obr. 3: Predposledný krok výpočtu vo variácii, kde máme povolený šikmý pohyb navyše.

Súťažná úloha

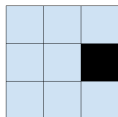
Hankina mama vymyslela ďalšie alternatívy úlohy, s ktorými si však Hanka už poradiť nevie. Jej postup je však veľmi flexibilný a myslí si, že by sa dal upraviť tak, aby riešil aj nasledovné varianty pôvodnej úlohy.

V popise napíšte, ako by ste riešili zadané problémy. Inšpirovať sa pritom môžete vyššie uvedeným postupom, nie je to však nevyhnutné. Ľubovoľné správne a dostatočne efektívne riešenie môže získať plný počet bodov. V popise sa sústreďte najmä na to, aby ste poriadne opísali ako vaše riešenie funguje a prečo je správne. Ak využívate Hankin postup, popíšte zmeny, ktoré robíte a dôvod, prečo sú potrebné a správne. Súčasťou popisu *nemusi* byť program.

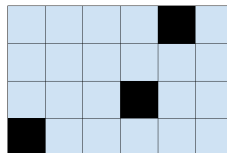
Pri každej podúlohe sú dva príklady toho, ako môže vyzeráť zadaná mriežka. K prvej ukážkovej mriežke je uvedené aj správne riešenie. Za vypočítanie správnej odpovede pre druhú zadanú mriežku môžete získať v každej podúlohe 1 bod. Váš popis však musí okrem výsledku obsahovať aj spôsob, akým ste túto hodnotu vypočítali.

- a) (2 body) Hanka má mriežku $r \times s$, v ktorej sa vie hýbať dodola a doprava. V mriežke sú navyše niektoré políčka neprechodné (označené čiernou), cez tieto políčka nemôže viesť žiadna cesta. Zistite, koľko rôznych ciest vedie z ľavého horného rohu do pravého dolného rohu.

dovolené
smery pohybu



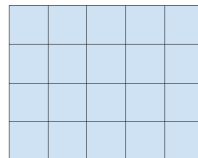
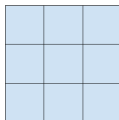
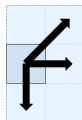
Správna odpoveď: 3





- b) (2 body) Hanka má mriežku $r \times s$. Hýbať sa však vie tromi spôsobmi – doprava, dodola a **šikmo doprava-hore**. Zistite, koľko rôznych ciest vedie z ľavého horného rohu do pravého dolného rohu.

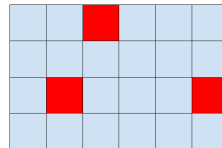
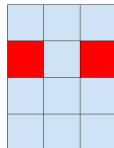
dovolené
smery pohybu



Správna odpoveď: 20

- c) (3 body) Hanka má mriežku $r \times s$, v ktorej sa vie hýbať dodola a doprava. V mriežke sú navyše niektoré políčka označené červenou farbou. Zistite, koľko rôznych ciest z ľavého horného rohu do pravého dolného rohu prechádza cez **aspoň jedno červené políčko**.

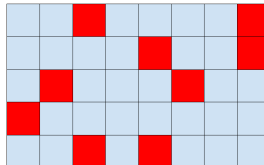
dovolené
smery pohybu



Správna odpoveď: 8

- d) (3 body) Hanka má mriežku $r \times s$, v ktorej sa vie hýbať dodola a doprava. V mriežke sú navyše niektoré políčka označené červenou farbou. Keď si zoberieme nejakú cestu z ľavého horného rohu do pravého dolného rohu, táto cesta prechádza cez niekoľko červených políčok (aj 0). Hankinu mamu zaujíma cesta, ktorá prechádza cez najväčší možný počet červených políčok. Takýchto ciest však môže byť viacero. Spočítajte počet rôznych ciest, ktoré prechádzajú cez najväčší možný počet červených políčok.

dovolené
smery pohybu



Správna odpoveď:
 najviac červených na ceste: 2
 takýchto ciest: 2

Odporúčanie: Niektorí z vás možno vedia, že pôvodná verzia úlohy (prázdna mriežka a dva druhy pohybov) sa dá vyriešiť aj pomocou kombinačných čísel. Použitie kombinačných čísel však nevedie k efektívnemu riešeniu nižšie uvedených súťažných podúloh. Odporúčame sa preto sústrediť na Hankin postup popísaný v zadání.



B-I-4 Krtko staviteľ

K tejto úlohe patrí študijný text uvedený na nasledujúcich stranách. Najskôr si prečítaj ten, až potom sa pusti do riešenia samotných súťažných úloh. Aj v nasledujúcich kolách tohto ročníka OI stretnes ten istý študijný text a úlohy, ktoré naň budú nadväzovať.

Táto úloha je teoretická. Odovzdať treba **jeden súbor vo formáte PDF** obsahujúci slovné popisy a zdôvodnenie správnosti jednotlivých podúloh.

V tomto ročníku olympiády sa stretneme s krtkami a budeme skúmať, aké nory si budujú.

Podúloha a)

Krtkovia majú veľmi radi hlinu. Nájdite sadu pravidiel pre krtka, ktorej portfólio $\mathcal{A}$ je tvorené iba norami, ktoré majú tvar $(\text{hlina})^n$, kde $n \geq 3$. Teda hlinahlinahlina , $\text{hlinahlinahlinahlina}$, ...

Podúloha b)

Nájdite sadu pravidiel, ktorej portfólio $\mathcal{B}$ je tvorené práve všetkými norami s nasledovnými dvoma vlastnosťami: každá miestnosť je typu **a** alebo **b**, pričom miestnosť typu **b** je v celej nore **práve jedna**. Teda do portfólia patria napr. **b**, **aaba**, ale nie **ε**, **bb** ani **sobota**.

Podúloha c)

Nájdite sadu pravidiel, ktorej portfólio $\mathcal{C}$ je tvorené práve všetkými norami zloženými z miestností **a** a **b**, v ktorých **nikdy nie sú** dve miestnosti typu **b** **vedľa seba**.

Teda napr. nory **abaab**, **ε**, **a²⁵⁶**, ale nie **abba**.

Podúloha d)

Nájdite sadu pravidiel, ktorej portfólio $\mathcal{D}$ je tvorené práve všetkými norami zloženými z miestností **a**, **b** a **c**, ktoré začínajú a končia miestnosťou rovnakého typu.

To spĺňajú napr. nory **abcbcacbca**, **c** alebo **ba⁸b**, ale nie nora **abc**.

Podúloha e)

Nájdite sadu pravidiel, ktorej portfólio $\mathcal{E}$ obsahuje tie nory, ktoré patria do $\mathcal{C}$ **a zároveň** aj do $\mathcal{D}$. Teda $\mathcal{E} = \mathcal{C} \cap \mathcal{D}$.

Hodnotenie

Za vyriešenie každej podúlohy viete získať 2 body. Jeden bod získate za nájdenie správnej sady pravidiel, ktorá bude mať popísané portfólio. Druhý bod viete získať za odôvodnenie, že vaša sada pravidiel je naozaj správna. V tomto odôvodnení by ste mali ukázať, že sada pravidiel vie vytvoriť ľubovoľnú zo zadaných nôr, a že ľubovoľnú inú noru postaviť nemôže.

Študijný text: Krtko staviteľ

Krtko rád kope nory. Krtia nora sa skladá z niekoľkých za sebou idúcich *dokončených* miestností a najviac jednej *nedokončenej* miestnosti na konci. Krtko potom každý deň príde do ešte nedokončenej miestnosti a prerobí ju na niekoľko (občas aj nula) dokončených miestností a najviac jednu nedokončenú miestnosť (ktorá, ak existuje, je opäť na konci). Keď sa krtia nora skladá už iba z dokončených miestností, považujeme ju za *dostavanú* a krtko na nej už ďalej nič nemení.

Existuje viacero typov dokončených aj nedokončených miestností. Dve nory rovnakej dĺžky sú rôzne, ak na niektorej pozícii majú noru iného typu.



Formálnejšia definícia

Krtia nora je postupnosť niekoľkých (možno žiadnych) miestností, pričom najviac jedna posledná miestnosť môže byť nedokončená. Dokončené miestnosti budeme označovať malými písmenami anglickej abecedy a číslicami; nedokončené miestnosti budeme označovať veľkými písmenami anglickej abecedy. Krtko teda pozná presne 36 typov dokončených miestností a 26 typov nedokončených miestností.

Prázdnu postupnosť písmen (teda 0 miestností v rade) budeme označovať ϵ (aby sme ju v texte videli). Ak máme v zápise za sebou k miestností a , môžeme ich skrátene zapísať ako a^k . Špeciálne $a^0 = \epsilon$.

Teda napríklad $a^{13}ba^{15}X$ je ešte nedostavaná krtia nora, ktorá má 29 dokončených a 1 nedokončenú miestnosť. aXb^3Yb nie je korektná nora, lebo obsahuje dve nedokončené miestnosti (X a Y). Nora $aaXb^2$ nie je korektná, lebo jediná nedokončená miestnosť nie je na konci postupnosti.

Na poradí a type miestností záleží, abc a cba sú dve rôzne dostavané nory.

Staviteľské pravidlo popisuje ako môže krtko svoju noru budovať. Každý krtko má vlastnú konečnú sadu pravidiel, ktorou sa riadi. Pravidlo popisuje, ako sa môže zmeniť *jedna nedokončená* miestnosť na nejakú postupnosť dokončených a najviac jednej nedokončenej miestnosti.

Napríklad $X \rightarrow abc$ ukazuje, že nedokončená miestnosť X sa môže zmeniť na uvedenú postupnosť troch dokončených miestností. Pravidlo $X \rightarrow Y$ dovoľuje zmeniť nedokončenú miestnosť typu X na nedokončenú miestnosť typu Y . A pravidlo $X \rightarrow cddcX$ môže zmeniť X na postupnosť $cddc$ dokončených miestností, na konci ktorých bude opäť nedokončená miestnosť X .

Povolené je aj použitie pravidla $X \rightarrow \epsilon$, ktoré nahradí nedokončenú miestnosť prázdnu postupnosťou, táto miestnosť teda v podstate zmizne.

Ak máme pre jednu nedokončenú miestnosť viacero pravidiel, môžeme ich navyše zjednodušene zapísať tak, že za šípkou uvedieme postupne všetky možnosti oddelené znakom $|$. Pre pravidlá z vyššie uvedeného príkladu by to vyzeralo nasledovne: $X \rightarrow abc | Y | cddcX | \epsilon$.

Vždy, keď má krtko zmeniť miestnosť X a existuje viacero pravidiel, ako zmeniť túto miestnosť, môže si vybrať a použiť ľubovoľné z nich. Ak má zmeniť miestnosť X a neexistuje žiadne pravidlo, ako zmeniť X , stavba nory sa zasekne – takáto nora teda nikdy nebude dostavaná.

Každá nora sa na začiatku skladá z jedinej nedokončenej miestnosti Z . Pre konkrétnu sadu pravidiel budeme slovom **portfólio** označovať množinu všetkých *dostavaných nôr*, ktoré dokáže krtko zo začiatku Z vybudovať postupnou aplikáciou zadaných pravidiel. (Upozorňujeme, že nory, v ktorých ešte máme nedokončenú miestnosť, do portfólia sady pravidiel nikdy nepatria.)

Majme sadu pravidiel $Z \rightarrow hline$. Portfóliom tejto sady pravidiel je množina obsahujúca jedinú noru, a to noru tvaru $hline$. Nič iné totiž krtko pomocou týchto pravidiel dostávať nevie.

Ak by sme mali dve pravidlá $Z \rightarrow \epsilon | aZ$, portfóliom tejto sady pravidiel sú všetky nory tvorené iba miestnosťami a . Zapísané formálne, sú to všetky nory tvaru a^n pre $n \geq 0$. Patria sem teda nory $\epsilon, a^1, a^2, a^3 \dots$. Všetky tieto nory sa naozaj postaviť dajú, pre zvolené n nám stačí n -krát použiť pravidlo $Z \rightarrow aZ$ a následne pravidlo $Z \rightarrow \epsilon$.

Nakoniec, pre sadu pravidiel $Z \rightarrow bcY | aY$ a $Y \rightarrow bZ | Z$ je portfólio prázdne. Takýmito pravidlami totiž nikdy žiadnu noru nedostávame, keďže po každej aplikácii niektorého pravidla nám na konci ostane nedokončená miestnosť.

Príklad 1

Zaujímá nás, či existuje taká sada pravidiel, ktorej portfólio je tvorené všetkými norami, ktoré sa skladajú iba z ľubovoľného počtu miestností a alebo b v ľubovoľnom poradí.

Takýmito norami sú napríklad nory $a, b, a^{600}, abbaa$ a nekonečne veľa ďalších kombinácií týchto dvoch znakov. (Vrátane nory ϵ .)

Riešenie

Hľadané pravidlá môžu vyzeráť nasledovne: $Z \rightarrow \epsilon | aZ | bZ$.



Je zjavné, že každá dostavaná nora bude pozostávať iba z miestností **a** a **b**. Potrebujeme však ešte ukázať, že každú takú noru dokáže krtko naozaj postaviť.

Kým krtkova nora nie je dostavaná, nutne bude končiť nedokončenou miestnosťou **Z** (keďže iné nedokončené miestnosti sa v pravidlách nenachádzajú). Zoberme si teraz nejakú noru tvorenú len z miestností **a** a **b**, napríklad noru **abbaab**. Krtko ju vie postaviť nasledovne: na stavbu použije druhé alebo tretie pravidlo podľa toho, ktorá miestnosť má byť v tejto nore na ďalšej pozícii. V našom prípade by teda použil druhé pravidlo, potom dvakrát tretie pravidlo, dvakrát opäť druhé pravidlo, a nakoniec tretie pravidlo. A keď už postaví všetky zadané miestnosti, aplikuje prvé pravidlo, čím noru dostavia, lebo odstráni nedokončenú miestnosť. Každá nora správneho tvaru teda patrí do portfólia týchto pravidiel.

Príklad 2

Existuje sada pravidiel, ktorej portfólio je tvorené dvoma norami – **hlina** a **a⁵⁵**?

Riešenie

Áno, napríklad sada pravidiel $Z \rightarrow Y$ a $Y \rightarrow \text{hlina} \mid a^{55}$. Krtko pri stavaní nemá veľmi na výber, nič okrem zadaných dvoch nôr nepostaví.

Určite ste si všimli, že v tejto sade je prvé pravidlo v podstate zbytočné a postačovalo by mať len jeden typ nedokončenej miestnosti a dve pravidlá: $Z \rightarrow \text{hlina} \mid a^{55}$. Samozrejme, máte pravdu. Rôzne sady pravidiel môžu mať rovnaké portfólio. Obe sady sú teda korektnými riešeniami nášho problému.

Príklad 3

Označme si portfólio z príkladu 1 ako $\mathcal{A}$ (teda nory, ktoré obsahujú ľubovoľnú kombináciu písmen **a** a **b**).

Označme si portfólio z príkladu 2 ako $\mathcal{B}$ (teda dvojica nôr **hlina** a **a⁵⁵**).

Existuje sada pravidiel, do ktorej portfólia $\mathcal{C}$ patria aj nory z $\mathcal{A}$ a zároveň aj nory z $\mathcal{B}$ (a žiadne iné)?
Formálne: Chceme, aby platilo $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$.

Riešenie

Pravidlá môžu byť nasledovné:

$$\begin{aligned} Z &\rightarrow A \mid B \\ A &\rightarrow \epsilon \mid aA \mid bA \\ B &\rightarrow \text{hlina} \mid a^{55} \end{aligned}$$

Túto sadu pravidiel sme vyrobili nasledovne:

- V sade pravidiel z príkladu 1 sme zmenili všetky výskyty nedokončenej miestnosti **Z** na **A**.
- V sade pravidiel z príkladu 2 sme zmenili všetky výskyty nedokončenej miestnosti **Z** na **B**.
- Zobrali sme obe tieto sady dokopy a navyše sme pridali pravidlá $Z \rightarrow A \mid B$.

Pri stavaní nory sa teda krtko v prvý deň rozhodne, či prerobí **Z** na **A** alebo na **B**. Ak vyrobí **A**, ďalej môže používať len pravidlá analogické príkladu 1, a ak vyrobí **B**, bude potom musieť postupovať rovnako ako v príklade 2.

Použitím prvého pravidla sa krtko rozhodne, či bude stavať noru z $\mathcal{A}$ alebo $\mathcal{B}$. Následne vie používať iba pravidlá z danej podúlohy. A aby sa tieto pravidlá navzájom neovplyvňovali, zmenili sme nedokončené miestnosti zo **Z** na dve unikátne nedokončené miestnosti **A** a **B**.

(Všimnite si ešte, že noru **a⁵⁵** vie vyrobiť aj prvým, aj druhým spôsobom.)

ŠTYRIDSIATY PRVÝ ROČNÍK OLYMPIÁDY V INFORMATIKE

Príprava úloh: Michal Anderle, Michal Farnbauer, Martin Belluš, Jozef Číž, Michal Forišek

Recenzia: Michal Anderle, Michal Forišek

Slovenská komisia Olympiády v informatike

Vydal: NIVAM – Národný inštitút vzdelávania a mládeže, Bratislava 2025