



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.

Termíny 42. ročníka OI

- Riešenia domáceho kola sa dá v kategórii A odovzdať do **15. 11. 2026**, v kategórii B do **30. 11. 2026**.
- Krajské kolo oboch kategórií prebehne 19. 1. 2027 zvlášť v každom kraji.
- Celoštátne kolo kategórie A sa bude konať v dňoch 17.-20. 3. 2027 (súťaží sa vo štvrtok a piatok).
- Detailnejšie informácie o priebehu súťaže nájdete na webe OI: <https://oi.sk/?s=pravidla>

Odovzdávanie riešení domáceho kola

Riešitelia domáceho kola odovzdávajú riešenia sami, v elektronickej podobe, a to priamo na stránke olympiády: <http://oi.sk/>. Na postup na krajské kolo je potrebné získať aspoň **10 bodov**.

Ako majú vyzeráť 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). Zoznam podporovaných jazykov vrátane konkrétnych verzií ich kompilátorov a interpreterov nájdete v systéme na odovzdávanie riešení. Tiež tam nájdete presnejší popis toho, ako majú vaše programy vyzeráť, 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 a zdôvodnenie jeho správnosti
- diskusiu o efektívnosti 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.

Používanie AI v rámci súťaže

Je **zakázané** využívať nástroje umelej inteligencie (AI), napr. veľké jazykové modely, na priame riešenie alebo implementáciu súťažných úloh. Všetky odovzdané výstupy (program aj slovný popis riešenia) musia byť plne dielom súťažiaceho.

Nástroje AI je možné využívať na štúdium a vysvetľovanie všeobecne známych algoritmov a programovacích techník. Napríklad, je v poriadku vyhľadať si pomocou AI informácie o „efektívnych algoritmoch na hľadanie najkratšej cesty“ a nechať si vysvetliť ich princíp alebo implementáciu. Nie je však dovolené použiť AI na hľadanie myšlienky ani na implementáciu riešenia konkrétnej súťažnej úlohy.

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.



A-I-1 Skriňa na chemikálie

Martin je vášnivý chemik. Celé popoludnia trávi v školskom chemickom labáku, kde robí najrôznejšie pokusy. Na pokusy však treba chemikálie a tie sú nanešťastie uskladnené v pivnici o dve poschodia nižšie, pociťovo tak milión schodov ďaleko.

Niektoré chemikálie si však Martin vie držať aj v skrini priamo v laboratóriu. Musí však dodržať jedno pravidlo – v skrini nemôžu byť naraz žiadne dve chemikálie, ktoré by spolu mohli *nevhodne reagovať*. Ak by niektorá fľaštička poriadne netesnila, bol by z toho v lepšom prípade smrad, v tom horšom nové laboratórium.

Každá chemikália má priradené evidenčné číslo. Platí, že dve chemikálie môžu nevhodne reagovať (a teda nemôžu byť spolu v skrini) práve vtedy, ak ich evidenčné čísla majú spoločného deliteľa väčšieho ako 1; teda ak sú súdeliteľné.

Ako Martin robí pokusy, nosí do laboratória nové chemikálie. Vždy, keď donesie novú látku, pozrie sa do skrine a skontroluje, či je bezpečné ju do nej umiestniť. Ak sa v skrini nenachádza žiadna chemikália, ktorej evidenčné číslo by bolo súdeliteľné s číslom novej chemikálie, Martin ju uloží do skrine. V opačnom prípade ju odnesie späť do pivnice. A občas, keď už nejakú chemikáliu viac nepotrebuje, vyberie ju zo skrine a tiež ju odnesie späť do pivnice.

Napište program, ktorý Martinovi pomôže rýchlo určiť, či môže zadanú chemikáliu bezpečne uložiť do skrine.

Súťažná úloha

Skriňa na chemikálie je na začiatku prázdna. Na vstupe dostanete postupnosť q udalostí dvoch typov:

1. **Vloženie** chemikálie s evidenčným číslom x do skrine. Ak sa v skrini nenachádza žiadna chemikália, ktorej číslo by bolo súdeliteľné s x , látka x je vložená do skrine a na výstup sa vypíše „**ano**“. V opačnom prípade sa x do skrine nevloží a na výstup sa vypíše „**nie**“.
2. **Vrátenie** chemikálie s číslom x do pivnice. Ak sa látka x nachádza v skrini, je odtiaľ vybraná. Ak nie, tak sa vôbec nič nestane. Na výstup sa pri tejto udalosti nič nevypisuje.

Formát vstupu a výstupu

Prvý riadok vstupu obsahuje číslo q – počet udalostí. Nasleduje q riadkov popisujúcich jednotlivé udalosti. Každá udalosť je popísaná dvoma číslami t_i a x_i . Hodnota t_i udáva typ udalosti a má buď hodnotu 1 alebo 2. Hodnota x_i udáva evidenčné číslo chemikálie spracovanej v tejto udalosti.

Pre každú udalosť s $t_i = 1$ vypíšte na výstup jeden riadok – „**ano**“ ak bola chemikália do skrine uložená, „**nie**“ ak sa tam bezpečne vložiť nedala.

Obmedzenia a hodnotenie

Pre všetky vstupy platí, že $1 \leq q \leq 2 \cdot 10^5$ a v každom vstupe pre každé i platí $t_i \in \{1, 2\}$ a $2 \leq x_i \leq 10^6$.

Testovacie vstupy sú rozdelené do štyroch sád. Pre jednotlivé sady platia nasledovné dodatočné obmedzenia:

sada	body	dodatočné obmedzenia
1	2	$q \leq 2000$
2	3	pre všetky i platí $x_i \leq 5000$
3	2	pre všetky i platí $t_i = 1$
4	3	bez dodatočných obmedzení



Príklady

vstup

```
6
1 6
1 10
2 6
1 10
1 15
1 7
```

výstup

```
ano
nie
ano
nie
ano
```

Chemikáliu 6 vložíme do prázdnej skrine. Chemikália 10 je s ňou súdeliteľná (spoločný deliteľ 2), preto sa vráti do pivnice. Počas tretej udalosti je skriňa opäť vyprázdnená, Martin tam teda následne môže počas štvrtej udalosti chemikáliu 10 vložiť. Chemikália 15 nevhodne reaguje s chemikáliou 10 (spoločný deliteľ 5), chemikália 7 však do skrine môže byť bez problémov uložená.

vstup

```
5
1 9
1 3
2 27
2 9
1 3
```

výstup

```
ano
nie
ano
```

Všimnite si, že na vstupe môžeme dostať požiadavku odnieť do pivnice chemikáliu, ktorá sa v skrini nenachádza, v tomto prípade číslo 27. V takom prípade sa nič nestane.

vstup

```
4
1 12
1 12
1 35
1 55
```

výstup

```
ano
nie
ano
nie
```

Druhé vloženie chemikálie 12 do skrine nebude úspešné, pretože 12 je súdeliteľná s 12, ktorá sa už v skrini nachádza.



A-I-2 Cap záhradníkom

Paulínka má doma v Lokci obrovitánsku kvetinovú záhradu. V záhrade sa nachádza n lavičiek, z ktorých ju môže obdivovať. Medzi lavičkami sa dá chodiť po vopred vyznačených *chodníkoch*, pozdĺž ktorých sú vysadené najrôznejšie kvety. Každý chodník spája práve dve lavičky a dá sa ním chodiť oboma smermi.

Keďže Paulínka je srdcom informatička, navrhla svoju záhradu tak, že tvorí *strom*. To znamená, že v záhrade je práve $n - 1$ chodníkov a platí, že medzi každými dvoma lavičkami existuje práve jedna postupnosť chodníkov (cesta), ktorou sa medzi nimi dá prejsť.

Niektoré chodníky však zarástli burinou a nie je na ne veľmi pekný pohľad. Paulínka preto každému chodníku priradila jedno celé číslo, ktoré označuje jeho estetickosť – chodníky s kladným hodnotením sú pekné, so záporným škaredé. Celkovú estetickosť záhrady potom Paulínka vypočíta ako súčet estetickostí všetkých chodníkov.

Aktuálny stav sa jej však nepáči a chcela by záhradu skrášliť. Zavolala preto do záhradníctva „Cap záhradníkom“, aby jej pomohli s údržbou. Konkrétne sa dohodli, že Paulínka určí jednu začiatočnú a jednu koncovú lavičku a zamestnanec firmy prejde po ceste medzi nimi a upraví výzor každého chodníka na nej.

Ako však názov firmy napovdá, výsledok údržby nie je vždy úplne ideálny. Škaredé chodníky síce vedľa vylepšiť, ale keď nájdú chodník pekný, tak ho svojím zásahom iba pokazia. Presnejšie, keď zamestnanec záhradníctva upraví chodník s estetickosťou e , bude mať tento chodník po úprave estetickosť $-e$.

Paulínka teraz rozmýšľa, akú najkrajšiu záhradu vie dosiahnuť zadaním jednej zákazky záhradníctvu.

Súťažná úloha

Máte zadaný popis záhrady, ktorú tvorí n lavičiek očíslovaných od 1 po n , medzi ktorými vedie $n - 1$ chodníkov. Každý chodník má na začiatku priradenú svoju estetickosť – jedno celé číslo.

Vašou úlohou je vybrať také dve lavičky, že keď zmeníme znamienko v estetickosti každého chodníka na ceste medzi týmito lavičkami, súčet estetickostí všetkých chodníkov v celej výslednej záhrade bude najväčší možný.

Je možné zvoliť rovnakú začiatočnú aj koncovú lavičku, v takom prípade nenastane žiadna zmena.

Formát vstupu a výstupu

Na prvom riadku vstupu je číslo n udávajúce počet lavičiek v záhrade.

Nasleduje $n - 1$ riadkov. Každý z nich obsahuje trojicu celých čísel a_i, b_i, e_i popisujúcu jeden chodník – medzi lavičkami s číslom a_i a b_i vedie chodník s estetickosťou e_i .

Je zaručené, že v každom vstupe záhrada tvorí strom.

Na výstup vypíšte jeden riadok a v ňom jedno celé číslo – najväčšiu možnú celkovú estetickosť záhrady po úprave jednej cesty v strome.

Obmedzenia a hodnotenie

Pre všetky vstupy platí, že $1 \leq n \leq 2 \cdot 10^5$. V každom vstupe pre každé i platí $-10^9 \leq e_i \leq 10^9$ a $1 \leq a_i, b_i \leq n$.

Testovacie vstupy sú rozdelené do piatich sád. Pre jednotlivé sady platia nasledovné dodatočné obmedzenia:

sada	body	dodatočné obmedzenia
1	1	$e_i > 0$ pre všetky i
2	2	$e_i < 0$ pre všetky i
3	2	$n \leq 100$
4	2	$n \leq 2000$
5	3	bez dodatočných obmedzení



Príklady

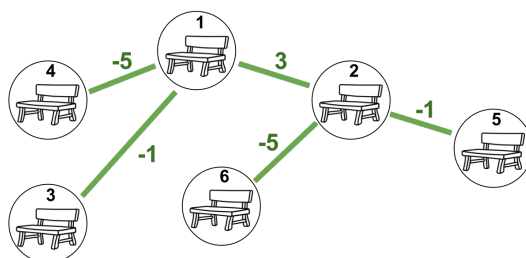
vstup

```
6
1 2 3
3 1 -1
4 1 -5
5 2 -1
6 2 -5
```

výstup

5

Na obrázku je zobrazené, ako vyzerá Paulínkina záhrada. Keď na úpravu zvolí cestu medzi lavičkami 4 a 6, záhradník prejde po chodníkoch s estetickosťou -5 , 3 a -5 a tieto hodnoty zmení na opačné. Celková estetickosť záhrady po úprave bude preto $(-3) + (-1) + 5 + (-1) + 5 = 5$.



vstup

```
5
1 2 1
1 3 6
1 4 7
4 5 3
```

výstup

17

Najlepšie je záhradu ponechať takú, aká je. Napríklad zvolením cesty medzi lavičkou 1 a 1.



A-I-3 Horské dráhy

Nedaleko Hornej Dolnej otvorili zábavný park. Jeho hlavnou atrakciou je sada horských dráh, po ktorých sa dá voziť.

V parku sa nachádza n staníc, tie sú očíslované od 1 po n . Medzi stanicami sú natáhané kolajnice pre jednotlivé úseky horskej dráhy. Úsekov je dokopy m a sú očíslované od 1 po m . Všetky úseky sú **jednosmerné**: úsek i vedie zo stanice x_i do stanice y_i .

Andrej sa rozhodol zobrať do tohto zábavného parku svoju malú dcérku Aničku.

V parku za k eur predávajú lístok „otec s dcérou kombo“, ktorý im dovolí na ľubovoľnej stanici nasadnúť na horskú dráhu a postupne sa ľubovoľne dlho voziť. Vždy, keď prídu do stanice, môžu si vybrať, či tam svoju jazdu ukončia, alebo prestúpia na niektorý z úsekov, ktorý tam začína a v jazde budú pokračovať.

Napríklad nech v parku existujú tri horské dráhy: zo stanice a do stanice b , z b do c a z c do a . Potom sa môžu naši hrdinovia na jeden lístok presunúť napr. z a do b , ale taktiež z a do c (s prestupom v b), alebo sa dokonca tri hodiny v kuse voziť medzi a , b a c dokola.

Lístkov vyššie uvedeného typu si môžu postupne kúpiť ľubovoľne veľa kusov.

Pozdĺž každého úseku horskej dráhy vedie aj chodník. Tieto chodníky sú **obojsmerné**. Anička je ochotná chodiť parkom aj pešo po týchto chodníkoch, ale len ak k tomu má cukrovú vatu. Takéto presuny teda Andreja budú stáť priamo úmerne tomu, ako dlhý je príslušný chodník – čím dlhší, tým viac toho Anička cestou zje. Peší presun po chodníku pozdĺž úseku i bude Andreja stáť d_i eur.

Súťažná úloha

Dostanete zadaný popis zábavného parku: počet staníc n , počet úsekov horskej dráhy m a cenu jedného lístka na ňu k . Pre každý úsek horskej dráhy dostanete číslo x_i stanice odkiaľ vedie, číslo y_i stanice kam vedie a cenu d_i za peší presun ľubovoľným smerom medzi týmito dvoma stanicami.

Andrej s Aničkou začínajú v stanici 1 a chcú sa dostať do stanice n . Navrhňte algoritmus, ktorý vypočíta, ako najlacnejšie toto vedia dosiahnuť.

Formát vstupu a výstupu

Prvý riadok vstupu obsahuje tri medzerou oddelené čísla n , m a k – počet staníc, počet úsekov horskej dráhy a cenu jedného lístka na ňu.

Nasleduje m riadkov. Na i -tom z nich je trojica čísel x_i , y_i a d_i . Tento riadok hovorí, že zo stanice x_i vedie úsek horskej dráhy na stanicu y_i a tiež to, že peší presun popri tomto úseku má pre Andreja cenu d_i .

Môžete predpokladať, že všetky usporiadané dvojice (x_i, y_i) sú navzájom rôzne a že pre každé i platí $x_i \neq y_i$.

Tiež môžete predpokladať, že hodnoty k a d_i (pre všetky i) sú kladné celé čísla nepresahujúce 10^9 .

Na výstup vypíšete jedno číslo – najmenšiu celkovú cenu presunu zo stanice 1 na stanicu n .

Obmedzenia a hodnotenie

Plný počet bodov viete získať za riešenie, ktoré efektívne vyrieši ľubovoľný vstup, kde $n, m \leq 10^5$. Jeho časová zložitosť by teda mala byť približne $O((n + m) \log m)$ alebo niečo tomu podobné.

Až 7 bodov môžete získať za riešenie, ktoré efektívne vyrieši ľubovoľný vstup, kde $n, m \leq 10^4$, teda zhruba so zložitostou $O(n \cdot m)$.

Až 5 bodov môžete získať za riešenie, ktoré predpokladá, že $n \leq 10^5$, $m = n - 1$ a úseky horskej dráhy sú vedené tak, že **chodníky popri nich** tvoria strom – teda pre každé dve stanice existuje práve jeden spôsob, ako sa po chodníkoch dostať z jednej do druhej. (Rozmyslite si, že v takejto situácii je vždy jednoznačne určené, cez ktoré stanice musí v optimálnom riešení Andrej postupne prechádzať. Všimnite si tiež, že vyššie uvedená podmienka nehovorí nič o tom, ktorým smerom je ktorý úsek horskej dráhy vedený – niektoré môžu ísť opačným smerom ako by sa Andrejovi hodilo.)

Za ľubovoľné správne riešenie, bez ohľadu na to ako je pomalé, viete získať aspoň 3 body.



Príklady

vstup

```
5 4 100
1 2 120
2 3 40
4 3 200
4 5 90
```

výstup

390

V tomto príklade existuje medzi stanicami 1 a $n = 5$ práve jedna cesta: Andrej s Aničkou budú postupne prechádzať cez stanice 2, 3 a 4 v tomto poradí. Optimálne riešenie je nasledovné: Andrej kúpi za 100 eur lístok a tým sa horskou dráhou odvezie zo stanice 1 cez 2 do stanice 3. Ďalej sa viezť nevedia, preto Andrej kúpi za 200 eur cukrovej vaty a presunú sa pešo do stanice 4. Tam by si vedeli kúpiť ďalší lístok za 100 a odviezť sa do cieľa, lacnejšie však je investovať ďalších 90 eur do cukrovej vaty a do stanice 5 prejsť opäť pešo. Celková cena tejto cesty je preto $100 + 200 + 90 = 390$.

vstup

```
9 8 50
1 8 10
5 1 70
2 5 80
5 3 30
3 6 20
3 9 40
9 4 60
7 9 60
```

výstup

120

Tento vstup tvorí strom. Optimálne je prejsť pešo zo stanice 1 do stanice 5 (70 eur) a odtiaľ sa na jeden lístok (50 eur) odviezť po trase $5 \rightarrow 3 \rightarrow 9$.

vstup

```
4 5 90
4 1 80
1 2 70
2 4 60
1 3 20
4 3 50
```

výstup

70



A-I-4 Optické hradlá

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

Podúloha A (1 bod): konštrukcia hradla NOT

Zostrojte optický počítač s dvoma emitormi a jedným kolektorom. Emitory budú označené 1 a x . Emitor 1 je vždy aktívny. Emitor x predstavuje vstup a môže byť buď aktívny alebo neaktívny. Váš počítač musí fungovať nasledovne: kolektor bude zasiahnutý lúčom práve vtedy, ak vstupný emitor **nebol** aktívny.

Okrem vyššie uvedených súčiastok smiete pri konštrukcii tohto počítača používať iba rozdeľovače a zrkadlá.

Podúloha B (2 body): konštrukcia hradla OR

Zostrojte optický počítač s tromi emitormi a jedným kolektorom. Okrem nich smiete pri konštrukcii tohto počítača používať iba rozdeľovače a zrkadlá. Emitory budú označené 1, x a y . Emitor 1 je vždy aktívny. Emitory x a y predstavujú vstup, každý z nich môže byť buď aktívny alebo neaktívny. Váš počítač musí fungovať nasledovne: kolektor bude zasiahnutý lúčom práve vtedy, ak **aspoň jeden** z emitrov x a y bol aktívny. Hodnota na výstupe je teda logickým OR hodnôt na vstupoch x a y .

Navyše v tejto podúlohe špeciálne platí, že **kolektor má vstup len zľava** – teda lúč, ktorý ho zasiahne, musí vždy priletieť zľava, bez ohľadu na to, ktorá kombinácia vstupov x a y bola aktívna.

Podúloha C (3 body): rytmické blikanie

Máme nasledovné parametre:

- Celé číslo n udávajúce počet výstupov.
- Kladné celé čísla $x_1, \dots, x_n$. Tieto čísla sú navyše všetky párne a väčšie ako 2.

Vašou úlohou je pre dané parametre zostrojiť optický počítač, ktorý bude fungovať nasledovne:

- Bude mať jeden emitor. Ten je aktívny.
- Bude mať n kolektorov, očíslovaných od 1 po n , v konfigurácii na priamke.
- Musí existovať nejaký čas t s nasledovnou vlastnosťou: pre každý kolektor i platí, že do neho laserový lúč narazí presne v časoch tvaru $t + kx_i$ pre nezáporné celé k .

Inými slovami, v čase t (ktorý si môžete zvoliť) musia laserové lúče po prvýkrát trafiť všetky kolektory (všetky naraz v tom istom okamihu) a následne bude každý kolektor i pravidelne znova a znova zasiahnutý laserom – vždy po x_i tikoch.

Podúloha D (4 body): súčet

Máme nasledovné parametre:

- Číslo n udávajúce počet vstupov.
- Číslo k udávajúce testovanú silu (vysvetlíme nižšie).
- Nezáporné celé čísla $s_1, \dots, s_n$: sily jednotlivých vstupov.

Vašou úlohou je pre dané parametre zostrojiť optický počítač, ktorý bude fungovať nasledovne:

- Bude mať n emitrov (očíslovaných od 1 po n) v konfigurácii na priamke. (Okrem nich môžete pri konštrukcii použiť aj ďalšie emitory, ktoré budú vždy aktívne.)
- Bude mať jeden kolektor.
- Celkovou silou vstupov nazveme súčet tých s_i , pre ktoré je emitor i aktívny.

Váš počítač by mal overiť, či je celková sila vstupov ostro menšia ako dané k . Kolektor by mal byť zasiahnutý laserom práve vtedy keď toto platí.



Študijný text: Optické hradlá

V tomto ročníku olympiády sa spolu pozrieme na jeden spôsob, akým sa dá na robenie výpočtov používať svetlo. Vydáme sa teda spolu do sveta plného laserových lúčov, odrazov a interferencie.

Základom každého nášho výpočtu bude *podložka*: obdĺžniková tabuľka, na ktorej je nakreslená štvorcová sieť. Každý štvorec tabuľky má rozmery 1×1 . Táto tabuľka je navyše vyfarbená šachovnicovým spôsobom – striedajú sa na nej teda čierne a biele polia. Optické počítače budeme zostrojovať tak, že si zoberieme dostatočne veľkú podložku a na niektoré jej políčka vhodne rozmiestnime súčiastky.

Medzi súčiastkami budú počas výpočtu lietať laserové lúče. Tieto budú mať len štyri možné smery – smer každého lúča bude rovnobežný s niektorou stranou podložky.

Typy súčiastok

V zozname uvedenom nižšie si predstavíme súčiastky, ktoré budeme používať pri stavbe počítačov.

Každá súčiastka má tvar jednotkového štvorca a musí byť umiestnená na niektoré políčko podložky. Na každom políčku podložky smie byť nanajvýš jedna súčiastka. Ak nie je povedané ináč, máme k dispozícii neobmedzené veľa kusov každej súčiastky.

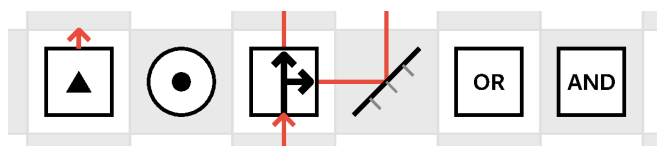
Niektoré súčiastky môžu mať na niektorých stranách vstupy a výstupy. V rámci návrhu počítača musíme vždy vyznačiť, kde presne sa tieto vstupy a výstupy nachádzajú. (Na každej strane každej takejto súčiastky sa môže nachádzať buď jeden vstup, alebo jeden výstup, alebo nič.)

- **Emitor:** Táto súčiastka nemá žiadne vstupy a má jeden výstup. Je to jediná súčiastka, ktorá vie vyrobiť nový laserový lúč. Každý emitor môže byť buď aktívny alebo neaktívny. V čase 0 každý aktívny emitor vystrelí na svoj výstup laserový lúč. Odvtedy už emitory nič nerobia.
To, ktoré emitory sú na začiatku výpočtu aktívne, budeme používať ako vstupné údaje pre výpočet.
- **Kolektor:** Táto súčiastka zachytáva laserové lúče. Toto budeme zväčša používať ako výstup výpočtu. Kolektor má vstupy zo všetkých strán a žiadny výstup. Ak do kolektora kedykoľvek vletí laserový lúč, kolektor to zaznamená.
- **Rozdeľovač:** Táto súčiastka sa dá použiť na duplikáciu lúčov. Má jeden vstup a ľubovoľne veľa výstupov (od nula po tri). Keď na vstup priletí laserový lúč, rozdelí sa a následne súčasne vyletí zo všetkých výstupov.
- **Zrkadlo:** Táto súčiastka otočí smer laserového lúča o 90° . Presnejšie, každé zrkadlo svojím natočením určuje dve susedné strany, cez ktoré odráža. Funguje tak, že keď cez jednu z nich priletí lúč, tak odletí cez druhú z nich.
- **OR:** Táto súčiastka robí logický or. Má ľubovoľne veľa vstupov (od jedna po tri) a jeden výstup. Z výstupu vyletí lúč práve vtedy, ak priletel lúč do aspoň jedného zo vstupov.
- **AND:** Táto súčiastka robí logický and. Má ľubovoľne veľa vstupov (od jedna po tri) a jeden výstup. Z výstupu vyletí lúč práve vtedy, ak súčasne priletel lúč do úplne všetkých vstupov.

Ak do nejakej súčiastky priletí lúč zo smeru, kde táto súčiastka nemá vstup, tento lúč okamžite zanikne.

Na obrázku nižšie sú schematické značky pre jednotlivé súčiastky, v poradí zodpovedajúcom vyššie uvedenému zoznamu. Červenou farbou sú naznačené smery laserových lúčov pre niektoré zo súčiastok, zodpovedajúce tomu, ako sú na obrázku natočené.

Všimnite si, že každý rozdeľovač má na sebe nakreslené, ktorá strana je vstup a ktoré strany sú výstupy. Toto môže mať každý rozdeľovač iné, pri stavbe počítača si to môžete zvoliť zvlášť pre každý z nich.





Konfigurácie vstupných a výstupných súčiastok

Pre **emitory** platí jedno obmedzenie, ktoré vždy musí byť dodržané: emitory smieme na podložke umiestňovať **len na biele políčka**.

V úlohách budeme spravidla používať dva typy rozmiestnení emitorov a kolektorov:

Pri **volnom rozmiestnení** si môžeme polohu a orientáciu príslušných súčiastok zvoliť ľubovoľne. Ak v zadaní nie je explicitne uvedené nič o rozmiestnení, ide o túto možnosť.

Pri **rozmiestnení na priamke** musia byť všetky príslušné súčiastky (emitory ak ide o vstup, kolektory ak o výstup) rozmiestnené na vodorovnej alebo zvislej priamke, a to v pravidelných rozstupoch. Ak sú takto rozmiestnené emitory, musia navyše byť všetky rovnako otočené (t.j. mať výstup tým istým smerom). Konkrétnu priamku aj veľkosť rozstupu medzi susednými súčiastkami si môžeme zvoliť.

Pohyb laserových lúčov

Každý laserový lúč má dĺžku 1 (teda rovnakú, ako strana políčka podložky). Čas, za ktorý sa laserový lúč posunie o jedno políčko, nazveme *tik*.

Všimnite si, že aj pri prelete cez všetky možné súčiastky majú laserové lúče rovnakú dĺžku dráhy ako pri prelete cez prázdne políčko. Je zaručené, že súčiastky lúče nijak nezrýchľujú ani nespomaľujú.

Ak sa niekedy laserové lúče zrazia, všetky okamžite zaniknú. To platí bez ohľadu na to, koľko lúčov sa zrazilo (aj ak sa zrazia viac ako dva naraz, všetky zaniknú) a tiež bez ohľadu na smer zrážky (či už si leteli naproti alebo kolmo na seba).

Pripomíname, že lúče zanikajú aj nárazom do strany súčiastky, kde nie je vstup.

Podotkneme ešte, že svetlo sa síce pohybuje spojitou, ale keďže každá zaujímavá udalosť trvá presne jeden tik, zväčša sa nič zlé nestane, ak o ňom budeme rozmýšľať ako o diskrétnom procese – teda ako keby výpočet prebiehal v krokoch, z ktorých každý trvá presne jeden tik.

Parametre, vstupy a výstupy

Vo všeobecnosti môžete očakávať, že riešenie konkrétnej úlohy na optickom počítači bude vyzeráť nasledovne:

Zadanie úlohy vám popíše, ako má konkrétny optický počítač fungovať – čo budú jeho vstupy, čo budeme považovať za jeho výstup a ako má tento výstup závisieť od vstupov. Zväčša budeme mať v počítači sadu emitorov, ktorých stavy na začiatku výpočtu (t.j. to, ktoré z nich sú aktívne) budeme považovať za vstup. Za výstup budeme zväčša považovať to, ktoré kolektory počas výpočtu zaznamenajú zásah laserovým lúčom.

Vašou úlohou bude samozrejme konštrukcia takéhoto optického počítača. Optický počítač, ktorý zostrojíte, musí byť *korektný* – pre každý možný vstup musí vyrobiť správny výstup. Ak nie je povedané ináč, môžete pri konštrukcii počítača použiť ľubovoľné súčiastky, a to **vrátane ďalších emitorov**. (Teda okrem emitorov, ktorých stavy predstavujú vstup, môže váš optický počítač použiť aj ďalšie. Tieto dodatočné emitory budú vždy aktívne.)

Niekedy bude stačiť priamo zostrojiť jeden konkrétny optický počítač. Toto bude platiť hlavne v situáciách, kedy je počet vstupov aj výstupov konštantný.

Inokedy však budeme chcieť vedieť spracovať vstupy alebo vyrábať výstupy ľubovoľnej veľkosti. Tu však narážame na drobný technický problém: každý optický počítač má nejaký konkrétny počet súčiastok. Tento problém vyriešime tak, že konštrukciu optického počítača budeme *parametrizovať*. Presnejšie:

1. Dostanete *parametre* optického počítača, ktorý máte zostrojiť.
Príkladom takéhoto parametra je číslo n udávajúce počet emitorov, ktoré má počítač obsahovať.
2. Pre dané parametre je potrebné algoritmicke zostrojiť konkrétny optický počítač. **Detaily tejto konštrukcie predstavujú samotné riešenie úlohy.** Vo všeobecnosti by ste preto mali v popise svojho riešenia uviesť:
 - Všeobecný algoritmus, ktorý pre dané parametre zostrojí konkrétny počítač.
 - Pre názornosť obrázku ukazujúce konštrukciu konkrétnych logických častí tohto počítača. (Jednotlivé logické časti budeme niekedy pre jednoduchosť nazývať *obvody*.)



(Formálne si tento krok môžete predstaviť tak, že v nejakom bežnom programovacom jazyku píšeme program, ktorý na vstupe dostane hodnoty parametrov, tie spracuje a následne na výstupe vypíše konkrétne rozmiestnenie súčiastok na podložke.)

3. Takto zostrojený optický počítač musí korektne fungovať pre ľubovoľný vstup, ktorý zodpovedá daným parametrom.

Toto celé si teraz ukážeme na dvoch príkladoch.

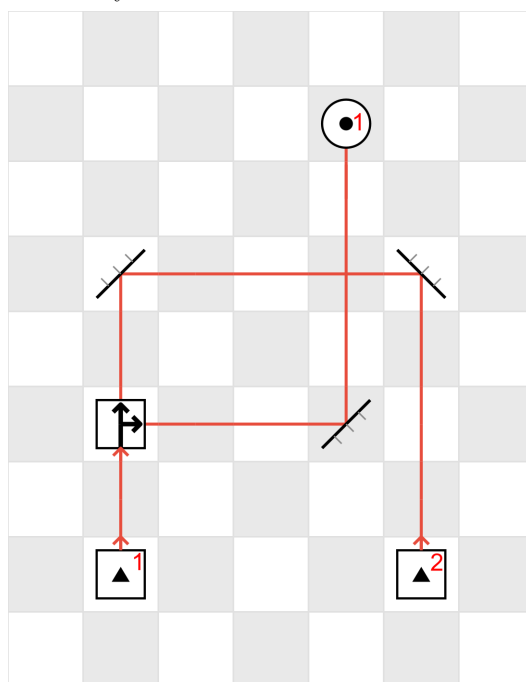
Príklad 1: konštrukcia súčiastky AND

V tomto príklade si ukážeme, že keby sme AND nemali k dispozícii ako súčiastku, vedeli by sme si postaviť obvod, ktorý sa bude správať rovnako ako táto súčiastka.

Zadanie: Zostrojte optický počítač s dvoma emitormi vo voľnej konfigurácii a jedným kolektorom. Vstupom pre jeho výpočet je to, ktoré emitory budú v čase 0 aktívne. Výstupom je to, či kolektor zachytil lúč alebo nie. Výstup počítača musí byť logickým ANDom oboch vstupov – teda kolektor musí zachytiť lúč vtedy a len vtedy, ak boli oba emitory aktívne.

(Táto úloha nemá žiadne parametre, zostrojujeme teda len jeden konkrétny počítač.)

Riešenie: Na obrázku nižšie je jedna možná konštrukcia takéhoto počítača, vrátane naznačených dráh, po ktorých poletia jednotlivé lúče, ak budú vystrelené z emitov.

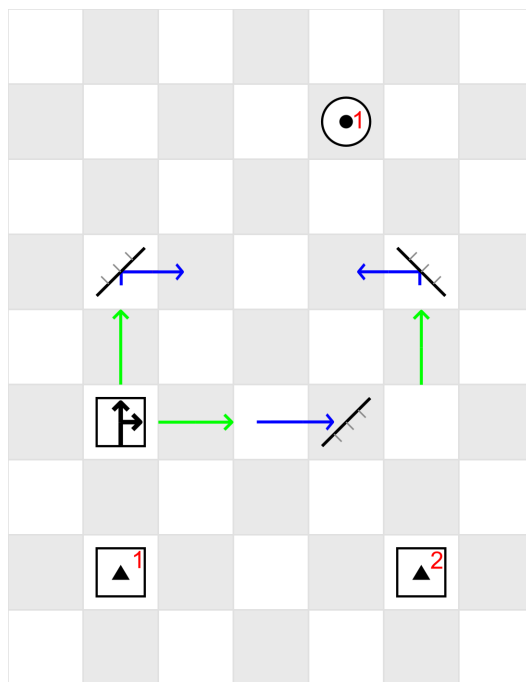


Rozoberme si jednotlivé možnosti pre vstupy:

- Ak nie je žiaden emitor aktívny, nemá čo do kolektora vletieť, čo je správne.
- Ak vystrelí lúč len emitör 1, ten sa v druhom tiku rozdelí. Následne sa v siedmom tiku tieto dva nové lúče zrazia, čím zaniknú. Do kolektora sa opäť nič nedostane.
- Ak vystrelí lúč len emitör 2, ten sa dvakrát odrazí a potom narazí do rozdeľovača zo strany, kde táto súčiastka nemá vstup. Tým tento lúč zanikne.
- Ak vystrelia lúče oba emitory, tak sa najskôr v druhom tiku lúč z emitora 1 rozdelí. Kópia idúca „hornou cestou“ sa následne v šiestom tiku zrazí s oproti letiacim lúčom z emitora 2. Tieto dva lúče teda zaniknú. Po siedmom tiku bude posledný existujúci lúč na políčku, cez ktoré predtým preletel lúč z emitora 2. Ten tam však už nie je, preto tento lúč spokojne pokračuje ďalej a v deviatom tiku trafi vstup kolektora.



Pre väčšiu názornosť ešte pridávame jeden obrázok ukazujúci, čo sa stane, ak sú aktívne oba emitory. Zelené šípky ukazujú polohu a smer všetkých laserových lúčov existujúcich po troch tikoch. Modré šípky zodpovedajú situácii niekedy počas piateho tiku (presnejšie, po približne 4.3 tikoch).



Príklad 2: vyhodnotenie postupnosti logických operácií

Táto úloha je príkladom úlohy s parametrami. V tejto konkrétnej úlohe budeme mať nasledovné parametre:

- Číslo n udávajúce počet logických premenných.
- Postupnosť $s_1, \dots, s_{n-1}$ logických operácií, pričom každé s_i je AND alebo OR.

Pre dané parametre je potrebné zostrojiť optický počítač, ktorý bude fungovať nasledovne:

- Počítač bude mať presne n emitov (očíslovaných od 1 po n) v konfigurácii na priamke.
- Emitor i bude zodpovedať logickej premennej x_i , ktorá hovorí, či bude aktívny.
- Počítač bude mať presne jeden kolektor. Výstupom bude to, či niekedy počas behu programu zachytil lúč.
- Výstup musí zodpovedať hodnote nasledovnej logickej formuly: $(\dots((x_1 s_1 x_2) s_2 x_3) \dots)$.

Teda napr. pre $n = 3$, $s_1 = \text{AND}$ a $s_2 = \text{OR}$ máme zostrojiť optický počítač, ktorý zo vstupných logických premenných x_1 , x_2 a x_3 na výstupe vypočíta hodnotu $((x_1 \text{ AND } x_2) \text{ OR } x_3)$.

Riešenie: Zvolíme si najmenší možný rozstup medzi emitormi, budú teda na každom druhom políčku. Umiestňovať ich budeme zľava doprava a všetky budú strieľať dohora.

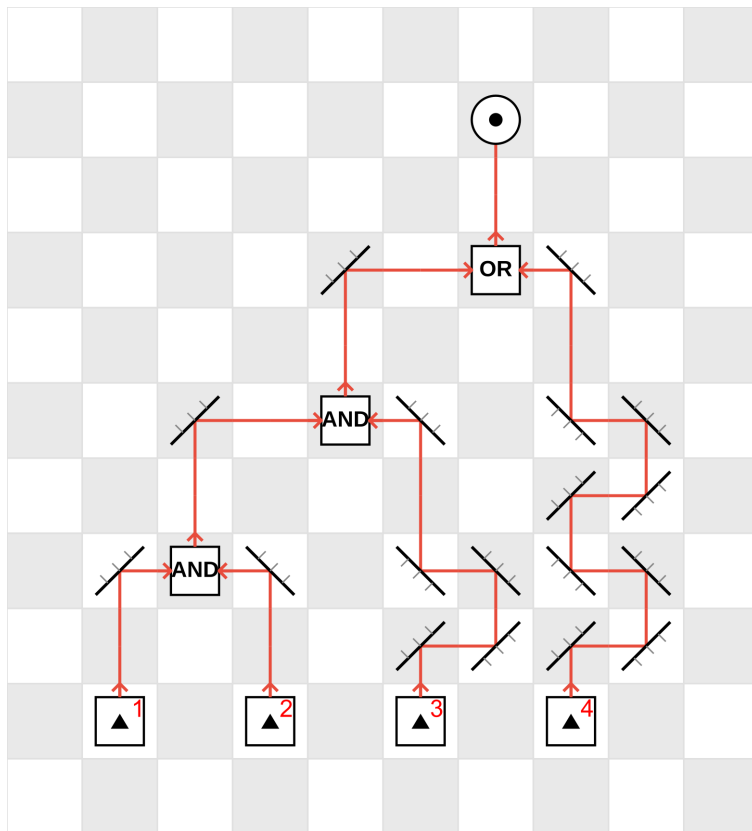
Pri samotnej konštrukcii takmer nebude záležať na jednotlivých parametroch s_i , tie ovplyvnia len to, či na príslušné políčko na podložke položíme súčiastku AND alebo OR.

Konštrukciu budeme robiť postupne zľava doprava. V i -tej iterácii pridáme obvod, do ktorého privedieme dva vstupy: prvým bude výstup doteraz zostrojenej časti počítača (pre $i = 1$ ide len o samotný emitov 1) a druhým bude výstup nasledujúceho emitora, teda emitora $i + 1$. Oba tieto vstupy privedieme do súčiastky, ktorá na nich spraví operáciu s_i . S výstupom z tejto súčiastky potom ďalej pracujeme v nasledujúcej iterácii, resp. po poslednej iterácii ho privedieme do kolektora.

Vo vyššie uvedenom popise konštrukcie nám ešte chýba jeden technický detail: každú súčiastku AND/OR musia oba potenciálne existujúce laserové lúče dosiahnuť **naraz**. Toto zabezpečíme tak, že v prípade potreby



jeden z lúčov spomalíme – presnejšie, navkladáme mu do cesty zrkadlá tak, aby mali oba lúče rovnako dlhú cestu k súčiastke, kde sa majú stretnúť.



Na obrázku vyššie je takto zostrojený optický počítač pre nasledovné parametre: máme $n = 4$ vstupy a postupnosť operácií, ktoré máme spraviť je $s_1 = \text{AND}$, $s_2 = \text{AND}$ a $s_3 = \text{OR}$.

Čitateľom odporúčame skontrolovať si, že pre každý AND aj OR naozaj platí, že ho lúče z oboch strán dosiahnu v tom istom okamihu. A ľahko tiež nahliadneme, že analogicky by sme vedeli pridávať aj ďalšie emitory a ďalšie logické spojky. Každý ďalší emitor by na spomalenie lúča dostal o 4 zrkadlá viac od toho predchádzajúceho.

ŠTYRIDSIATY DRUHÝ ROČNÍK OLYMPIÁDY V INFORMATIKE

Príprava úloh: Michal Anderle, Michal Forišek

Recenzia: Michal Forišek

Slovenská komisia Olympiády v informatike

Vydal: NIVAM – Národný inštitút vzdelávania a mládeže, Bratislava 2026