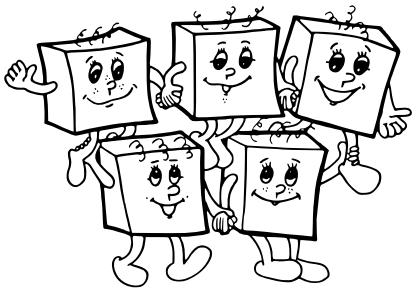
OLYMPIÁDA V INFORMATIKE

NA STREDNÝCH ŠKOLÁCH

http://oi.sk/



dvadsiaty ôsmy ročník
školský rok 2012/2013

zadania krajského kola
kategória B

Priebeh krajského kola

Krajské kolo 28. ročníka Olympiády v informatike, kategória B, sa koná 22. januára 2013 v dopoludňajších hodinách. Na riešenie úloh majú súťažiaci **4 hodiny čistého času**. Rôzne úlohy riešia súťažiaci na samostatné listy papiera. Akékoľvek pomôcky okrem písacích potrieb (napr. knihy, výpisy programov, kalkulačky) sú zakázané.

Čo má obsahovať riešenie úlohy?

- Slovne popíšte algoritmus.
  Slovný popis riešenia musí byť jasný a zrozumiteľný i bez nahliadnutia do samotného algoritmu/programu.
- Zdôvodnite správnosť vášho algoritmu.
- Uveďte a zdôvodnite jeho časovú a pamäťovú zložitosť.
- Podrobne uveďte dôležité časti algoritmu, ideálne vo forme programu v Pascale alebo C/C++.
- V prípade, že používate vo svojom programovacom jazyku knižnice, ktoré obsahujú implementované dátové štruktúry a algoritmy (napr. STL pre C++), v popise algoritmu stručne vysvetlite, ako by ste napísali program s rovnakou časovou zložitosťou bez použitia knižnice.

Hodnotenie riešení

Za každú úlohu môžete získať od 0 do 10 bodov.

Pokiaľ nie je v zadaní povedané ináč, najdôležitejšie dve kritériá hodnotenia sú v prvom rade **správnosť** a v druhom rade **efektívnosť** navrhnutého algoritmu. Na výslednom počte bodov sa môže prejaviť aj kvalita popisu riešenia a zdôvodnenie tvrdení o jeho správnosti a efektívnosti.

Efektívnosť algoritmu posudzujeme vypočítaním jeho časovej zložitosti – funkcie, ktorá hovorí, ako dlho vykonanie algoritmu trvá v závislosti od veľkosti vstupných parametrov. Nezávisí pri tom na konštantných faktoroch, len na rádovej rýchlosti rastu tejto funkcie.

V zadaní úloh uvádzame časť „Hodnotenie“, v ktorej nájdete približné limity na veľkosť vstupných údajov. Pod pojmom „efektívne vyriešiť“ chápeme to, že váš program spustený na modernom počítači by mal dať odpoveď nanajvýš do niekoľkých sekúnd.

Údaje z tejto časti zadania by mali slúžiť hlavne na to, aby ste o riešení, ktoré vymyslíte, vedeli približne povedať, koľko bodov zaň dostanete.

B-II-1 Nutela

Luxusko si kupuje zásoby nutely na dnešný večer. Stojí pred dlhou policou v potravinách, kde sú v rade naukladané nutely v rôznych téglikoch. Tégliky nutely sú na polici zoradené vzostupne podľa veľkosti. Úplne naľavo je najmenšie balenie nutely a napravo najväčšie.

Dnes majú v potravinách akciu: ak si zákazník kúpi práve dve nutely, dostane k nim zadarmo lyžičku. Luxusko už štvrť hodiny stojí a rozmýšľa, ktoré dve nutely si zobrať. Je dôležité, aby sa dobre najedol, takže nemôže kúpiť príliš málo nutely. Ale zase nechce príliš pribrať, takže jej nesmie kúpiť ani priveľa.

Súťažná úloha

Vašou úlohou je napísať program, ktorý poradí Luxuskovi, ktoré dve nutely má kúpiť, aby súčet ich veľkostí bol aspoň a, ale nie viac než b. Nutely majú byť rôzne – nemôže kúpiť dvakrát tu istú nutelu.

Ak žiadna dvojica nutiel nevyhovuje, vypíšte o tom správu.

Formát vstupu

V prvom riadku vstupu máte tri kladné celé čísla – počet téglikov na polici n, dolnú hranicu a a hornú hranicu b. Môžete predpokladať, že $a \leq b$.

V druhom riadku je n **rôznych** kladných celých čísel **usporiadaných vzostupne**, pričom i-te z nich vyjadruje veľkosť i-tej nutely na polici.

Veľkosti jednotlivých nutiel aj čísla a a b môžu byť pomerne veľké (napr. rádovo do 10^{18}), pri písaní programu však môžete predpokladať, že sa zmestia do bežných celočíselných premenných.

Formát výstupu

Ak existujú nejaké dve **rôzne nutely**, ktorých súčet veľkostí je medzi a a b vrátane, vypíšte veľkosti týchto nutiel. V prípade, že je možností viac, vypíšte ľubovoľnú jednu z nich.

Inak vypíšte jeden riadok so správou „nedá sa“.

Hodnotenie

Plných 10 bodov môžete dostať len za riešenie, ktorého (asymptotická) časová zložitosť je optimálna.
Aspoň 8 bodov môže dostať ľubovoľné riešenie, ktoré efektívne vyrieši ľubovoľný vstup s $n \leq 100\,000$.
Na zisk 6 bodov stačí úlohu efektívne vyriešiť za predpokladov $n \leq 100\,000$ a $a = b$.
Na zisk 5 bodov stačí úlohu efektívne vyriešiť za predpokladov $n \leq 100\,000$, $a = b$ a $b \leq 100\,000$.
Ľubovoľné pomalé ale korektné riešenie môže získať až 3 body.

Príklad

vstup

```
5 10 12
1 2 4 7 11
```

výstup

```
4 7
```

Tento vstup má práve dve správne riešenia. Druhým je dvojica 1 + 11.

vstup

```
4 10 100
3 4 5 123456
```

výstup

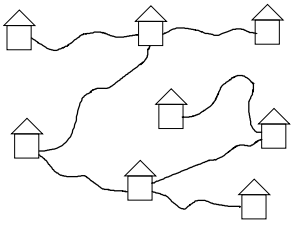
```
nedá sa
```

V tomto vstupe má každá dvojica nutiel buď primalý alebo priveľký súčet.

B-II-2 Pošta

Jano vlastní poštovú spoločnosť J&P. Nedávno sa rozhodol zaviesť novinku: začne ponúkať aj expresné doručovanie. To bude fungovať tak, že človek si priamo do domu zavolá poštára, dá mu list, a poštár ho hneď odnesie na zadanú adresu. Aby Jano vedel, či takáto služba bude komerčne úspešná, rozhodol sa ju najskôr otestovať v malebnej slovenskej dedine Oklieštené Ihličnany.

Oklieštené Ihličnany sú vcelku nezvyčajná dedina. Skladá sa z n domov a $n-1$ ulíc, pričom každá ulica priamo spája niektoré dva domy. Až na začiatky a konce sa ulice nikde nekrižujú. Po každej ulici sa dá chodiť oboma smermi. Ulice sú navrhnuté tak, aby sa po nich dalo dostať od každého domu ku každému inému. (Odborne povedané, z vyššie uvedených informácií vyplýva, že dedina má stromovú topológiu.)



Príklad dediny a ulíc v nej.

Nová poštová služba sa od prvej chvíle stala veľmi obľúbenou, však v takej dedine je plno noviniek, o ktoré sa chcú jej obyvatelia podeliť. Preto sa stalo, že hneď prvý deň musel Jano postupne jednu po druhej odniesť $n(n-1)$ zásielok: z každého domu do každého iného domu. Teraz by ho zaujímalo, koľko toho za dnešok s listom v ruke nachodil.

Súťažná úloha

Na vstupe dostanete popis dediny Oklieštené Ihličnany. Vašou úlohou bude spočítať súčet dĺžok ciest medzi všetkými možnými dvojicami domov. Pod dĺžkou cesty medzi domami x a y rozumieme počet ulíc, ktoré musí Jano prejsť, aby sa z domu x dostal do domu y. (Uvedomte si, že pre každú dvojicu x a y je postupnosť ulíc, ktorými treba ísť, jednoznačne určená, ak nechceme ísť tou istou ulicou tam aj späť.)

Formát vstupu a výstupu

V prvom riadku vstupu je číslo n určujúce počet domov. Domy sú očíslované od 1 po n. Zvyšok vstupu tvorí $n-1$ riadkov popisujúce ulice. Popis ulice sú dve čísla x, y, ktoré znamenajú, že medzi domami x a y vedie ulica. Je zaručené, že ulice tvoria jeden súvislý strom.

Na výstup vypíšte jedno číslo, súčet dĺžok ciest medzi každými dvoma domami.

Hodnotenie

Plných 10 bodov dostanete za riešenie, ktoré efektívne vyrieši ľubovoľný vstup s $n \leq 1\,000\,000$.
Až 7 bodov môžete získať za riešenie, ktoré efektívne vyrieši ľubovoľný vstup s $n \leq 5\,000$.
Až 4 body môžete získať za riešenie, ktoré efektívne vyrieši ľubovoľný vstup s $n \leq 300$.

Príklad

vstup

```
4
1 2
3 1
1 4
```

výstup

```
9
```

Máme 6 dvojíc domov. Medzi domami 1 a 2 vedie priama ulica, ich vzdialenosť je teda 1. Podobne majú vzdialenosť 1 aj dvojice (1,3) a (1,4). Pre každú z dvojíc (2,3), (2,4) a (3,4) je dĺžka cesty 2. Súčet všetkých týchto vzdialeností je 1+1+1+2+2+2 = 9.

vstup

```
5
1 2
1 3
2 4
2 5
```

výstup

```
18
```

Spomedzi 10 dvojíc domov majú štyri dvojice vzdialenosť 1, štyri dvojice vzdialenosť 2 a dve dvojice vzdialenosť 3.

B-II-3 Poklad

Keď Georg jedného dňa vysával dlhočiznú rovnú chodbu vo svojej vile, všimol si, že jeho vysávač zvláštne hrká. Zrazu sa objavil duch jeho (pra)[47]dedka a vyjavil mu, že chodba prechádza miestom, kde pred stovkami rokov zakopali obrovský poklad. Vysávač hrká kvôli tomu, že naň poklad elektromagneticky pôsobí. Duch síce neprezradil, kde presne je pod chodbou poklad ukrytý, ale Georg dostal nápad: zistí to z hrkania vysávača. Čím viac vysávač hrká, tým musí byť poklad bližšie.

Georgov sluch ani pamäť však nestačia na to, aby z hrkania na nejakom mieste priamo povedal, kde sa poklad nachádza. Nanajvýš tak dokáže rozpoznať to, či teraz hrká vysávač viac alebo menej ako na predchádzajúcom vysávanom políčku.

Chodba sa skladá z n políčok uložených v rade za sebou. Políčka sú očíslované zaradom od 1 po n. Poklad leží pod jedným z nich. Georgovi záleží na životnom prostredí a šetrí elektrinou, takže by chcel pri hľadaní pokladu povysávať čo najmenej políčok. (Navyše, čo ak by bol účet za elektrinu vyšší ako cena nájdeného pokladu?)

Súťažná úloha

Georg chce program, ktorý mu bude radiť, v akom poradí má pri hľadaní pokladu políčka vysávať. Najdôležitejšie je, aby váš program vždy poklad našiel. Navyše sa snažte, aby mal čo najmenšiu spotrebu elektriny – teda aby počet Georgom povysávaných políčok bol (v najhoršom možnom prípade) čo najmenší. Úloha má dve verzie, každú z nich riešte samostatne.

a) (5 bodov) Georg má predlžovačku, vďaka ktorej môže vysávať aj políčka mimo chodby. Inými slovami, číslo vysávaného políčka môže byť ľubovoľné celé číslo. (Poklad je ale určite pod jedným z políčok 1 až n.) Pre dĺžku chodby platí $2 \leq n \leq 10^9$. Na plný počet bodov nesmie váš program nikdy potrebovať povysávať viac ako 35 políčok.

b) (5 bodov) Georg predlžovačku nemá, a teda môže vysávať iba políčka na chodbe. Inými slovami, číslo každého vysávaného políčka musí byť celé číslo z rozsahu od 1 do n. Plný počet bodov dostanete, ak spotreba vášho programu bude vždy najviac dvojnásobkom najhoršej možnej spotreby optimálneho programu. Teda ak optimálnemu programu vždy stačí povysávať najviac x políčok, vám musí stačiť $2x$.

Formát vstupu a výstupu

Váš program bude s Georgom komunikovať pomocou štandardného vstupu a výstupu. Na začiatku Georg programu oznámi číslo n, udávajúce počet políčok chodby. Potom bude program bežať v cykle: Na výstup vypíše Georgovi, ktoré políčko má povysávať, a zo vstupu následne načíta Georgovu odpoveď. Tou bude jeden z reťazcov „`blizsie`“, „`dalej`“ alebo „`rovnako`“, podľa toho, či je práve vysávané políčko k pokladu bližšie, od neho ďalej, alebo rovnako ďaleko ako predchádzajúce vysávané políčko. (Pre prvé vysávané políčko mu Georg odpovie „`neviem`“.) Keď už si je program istý polohou pokladu, namiesto vysávania pošle Georga kopať.

Príklad

vstup

```
10
neviem
blizsie
rovnako
```

výstup

```
vysavaj 2
vysavaj 3
vysavaj 9
kop 6
```

Udiali sa nasledovné udalosti:

- *Program poslal Georga povysávať políčko 2.*
- *Georg povysával políčko 2 a odpovedal, že o poklade nič nevie.*
- *Program poslal Georga povysávať políčko 3.*
- *Georg povysával políčko 3 a odpovedal, že políčko 3 je k pokladu bližšie ako políčko 2.*
- *Program poslal Georga povysávať políčko 9.*
- *Georg povysával políčko 9 a odpovedal, že políčko 9 je od pokladu rovnako ďaleko ako políčko 3.*

Z toho už program usúdil, že poklad musí byť pod políčkom 6.

B-II-4 Roboti

Na číselnej osi žijú maličkí roboti. Číselnú os majú rozdelenú na políčka dĺžky 1 cm a sami sa pohybujú zásadne s krokom tejto dĺžky. Každým krokom sa teda robot dostane o jedno políčko doľava alebo doprava.

Na každom políčku môže byť (najviac jeden) kamienok. Robot vie kamienky na políčka klásť, zase ich zbierať aj zistiť, či je na políčku kamienok. Kamienky mu v pohybe nijak neprekážajú. Na kladenie ich má pripravených dostatočne veľa, takže môžete predpokladať, že sa mu nikdy neminú. Ak je na číselnej osi viac robotov, nijak si neprekážajú a vidia sa, ak sú na tom istom políčku.

Robotov môžeme programovať. Program pre robota je postupnosť inštrukcií, očíslovaných rastúcimi prirodzenými číslami. Každú inštrukciu robot vykoná za 1 sekundu. Ak sa robotovi minú inštrukcie, sám zastane (ako keby za poslednou inštrukciou bola ešte inštrukcia `koniec`). Roboti poznajú nasledovné inštrukcie:

`koniec`	skonči vykonávanie programu
`nic`	jednu sekundu nerob nič `:-)`
`vlavo`, `vpravo`	pohni sa o políčko doľava, resp. doprava
`zdvihni`, `poloz`	zdvihni kamienok, ak tu nejaký je / polož kamienok, ak tu ešte nie je
`pokracuj x`	vo vykonávaní programu pokračuj inštrukciou x.
`ak je kamienok, pokracuj x`	ak je na tvojom políčku kamienok, pokračuj inštrukciou x.
`ak je robot, pokracuj x`	ak je na tvojom políčku iný robot, pokračuj inštrukciou x.

Príklad

Robot začína na nejakom prázdnom políčku na číselnej osi. Napravo od neho sú na niektorých políčkach rozmiestnené (dokopy aspoň 2) kamienky. Po spustení nasledujúceho programu robot nájde druhý kamienok napravo od neho, zodvihne ho a na danom políčku zastane. (Pre úsporu miesta je program v dvoch stĺpcoch.)

```
10: ak je kamienok, pokracuj 40
20: vpravo
30: pokracuj 10
40: vpravo
```

```
 50: ak je kamienok, pokracuj 999
 60: vpravo
 70: pokracuj 50
999: zdvihni
```

Súťažná úloha

a) (2 body) Robot začína na nejakom prázdnom políčku na číselnej osi. Aj naľavo, aj napravo od neho je na práve jednom z políčok kamienok. Napíšte program, po ktorého spustení bude robot dokola behať od jedného kamienku k druhému a späť. (Tento program má bežať do nekonečna.)

b) (3 alebo 4 body) Robot začína na nejakom prázdnom políčku na číselnej osi. Aj naľavo, aj napravo od neho je na práve jednom z políčok kamienok. Vzdialenosť medzi kamienkami je párna, teda existuje nejaké políčko p, ktoré je presne uprostred medzi políčkami označenými kamienkom.

Tri body dostanete za program, po ktorého spustení robot nájde políčko p a zastane na ňom. (Nezáleží na časovej zložitosti programu, ani na tom, kde budú a kde nebudú na číselnej osi kamienky.) Štvrtý bod dostanete za to, ak počas behu vášho programu robot číselnú os uprace: v okamihu, keď na políčku p váš robot ukončí svoj program, nesmie byť na číselnej osi nikde ani jeden kamienok.

c) (4 body) Na číselnej osi nie sú nikde žiadne kamienky. Zobrali sme dvoch robotov a položili ich na dve rôzne políčka. Napíšte *jeden program*, ktorý nahráme do *oboch robotov* a naraz spustíme. Cieľom programu je, aby sa obaja roboti stretli na jednom políčku (je jedno, ktorom) a ukončili tam svoj program.

Všimnite si, že pri písaní programu neviete, ako ďaleko od seba roboti začínajú. Zdôrazňujeme, že obaja roboti musia dostať *presne rovnaký* program, ktorý potom budú synchrónne krok za krokom vykonávať.

DVADSIATY ÔSMY ROČNÍK OLYMPIÁDY V INFORMATIKE

Príprava úloh: Michal Anderle, Tomáš Belan, Michal Forišek, Ján Hozza
Recenzia: Michal Forišek
Slovenská komisia Olympiády v informatike
Vydal: IUVENTA – Slovenský inštitút mládeže, Bratislava 2012