



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 niekedy v septembri.

Riešenia kategórie A je potrebné odovzdať najneskôr **15. novembra 2022**.

Riešenia kategórie B je potrebné odovzdať najneskôr **30. novembra 2022**.

Priebeh súťaže

Za každú úlohu domáceho kola sa dá získať od 0 do 10 bodov. Na základe bodov domáceho kola stanoví Slovenská komisia OI (SK OI) pre každú kategóriu bodovú hranicu potrebnú na postup do **krajského kola**. Očakávame, že táto hranica bude približne rovná **tretine maximálneho počtu bodov**.

V krajskom kole riešitelia riešia štyri teoretické úlohy, ktoré môžu tematicky nadväzovať na úlohy domáceho kola. V kategórii B súťaž týmto kolom končí.

V kategórii A je približne najlepších 30 riešiteľov krajského kola (podľa počtu bodov, bez ohľadu na kraj, v ktorom súťažili) pozvaných do **celoštátneho kola**. V celoštátnom kole účastníci prvý deň riešia teoretické a druhý deň praktické úlohy. Najlepší riešitelia sú vyhlásení za víťazov. Približne desať najlepších riešiteľov následne SK OI pozve na týždňové výberové sústreďenie. Podľa jeho výsledkov SK OI vyberie družstvá pre Medzinárodnú olympiádu v informatike (IOI) a Stredoeurópsku olympiádu v informatike (CEOI).

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++, Pascal alebo Java). 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.

Presný popis, ako majú vyzerat' riešenia praktických úloh (napr. realizáciu vstupu a výstupu), nájdete na webstránke, kde ich budete odovzdávať.

Ak nie je v zadaní povedané ináč, riešenia teoretických úloh musia v prvom rade obsahovať **podrobný slovný popis použitého algoritmu, zdôvodnenie jeho správnosti** a diskusiu o efektivite zvoleného riešenia (t. j. posúdenie časových a pamäťových nárokov programu). Na záver riešenia uveďte program. Ak používate v programe netriviálne algoritmy alebo dátové štruktúry (napr. rôzne súčasti STL v C++), súčasťou popisu algoritmu musí byť dostatočný 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 IUVENTA v tesnej súčinnosti so Slovenskou komisiou OI.



B-I-1 Syroválač

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.

Francúzsko je krásna krajina, okrem iného hrdá na svoje syry. Keď Jozef toto leto navštívil francúzske hory, všimol si, že na vidieku sú len dva typy domov: tie, v ktorých každý deň jedno koleso syru určitého typu **vyrobia**, a tie, v ktorých každý deň jedno koleso syru určitého typu **skonzumujú**.

Nastáva teda pomerne náročný logistický problém – od koho si kto má kúpiť syr tak, aby boli všetci konzumenti spokojní. To je navyše skomplikované tým, v že horách existuje len jeden spôsob akým prepravovať kolesá syru – skotúľať ich dole kopcom.

Jozef hneď zacítil príležitosť pre biznis. Mohol by pomôcť celej dedinke v horách zoptimalizovať množstvo kotúľania, ktoré musia spraviť, aby každý dostal svoj syr. Vytvoril teda aplikáciu Syroválač. Dizajn, reklamnú kampaň aj testovacie dáta už má a tú ťažšiu časť, zostrojenie algoritmu na riešenie problému, nechal na vás.

Súťažná úloha

Na vstupe sa nachádza popis jednej ulice. Ulica ide dole kopcom a stojí na nej n domov. Každé dva susedné domy sú od seba vzdialené 1 meter. Každý dom buď nejaký typ syra vyrába alebo nejaký typ syra konzumuje. Zistite, koľko najmenej metrov musia dokopy na tejto ulici kotúľať každý deň syry tak, aby boli uspokojené všetky domy, v ktorých syr konzumujú.

Domy na ulici budú zadané vždy v poradí, v ktorom sú na ulici, teda od najvyššie položeného po najnižšie položený. Ak máte dom vyrábajúci syr typu x na pozícii i a dom, ktorý konzumuje syr typu x na pozícii j takej, že $j > i$, tak dom i môže poslať syr domu j . Takto poslaný syr budú musieť kotúľať $j - i$ metrov.

Nezabudnite, že každý výrobca môže poslať **najviac jeden** syr a každý konzument chce dostať **práve jeden** syr. Pri riešení môžete predpokladať, že existuje aspoň jeden spôsob, akým vyhovieť všetkým konzumentom.

Formát vstupu a výstupu

Vstup obsahuje jeden reťazec znakov dĺžky n popisujúci ulicu. Reťazec sa skladá z malých a veľkých písmen anglickej abecedy. Veľké písmeno na i -tej pozícii symbolizuje, že i -ty dom vyrába syr tohto typu a malé písmeno, že daný dom by syr tohto typu rád kúpil. (Např. X je dom vyrábajúci syr typu x a x je dom, ktorý syr tohto typu chce.)

Na výstup vypíšte jedno číslo nasledované znakom konca riadku. Toto číslo má povedať, koľko najmenej metrov kotúľania je potrebných na presunutie syrov od výrobcov ku konzumentom za predpokladov uvedených vyššie.

Obmedzenia a hodnotenie

Táto úloha má 5 hodnotených vstupov. Pre dĺžku ulice a teda aj pre dĺžku reťazca na vstupe vždy platí, že $n \leq 3\,000\,000$. Okrem toho pre niektoré vstupy platia obmedzenia navyše:

Pre ulicu v prvom vstupe platí, že je krátka ($n \leq 1\,000$) a zároveň na tejto ulici poznajú len jeden druh syra.

Pre ulicu v druhom vstupe platí, že je krátka ($n \leq 1\,000$).

Pre ulicu v treťom vstupe platí, že na nej poznajú len jeden druh syra.



Príklady

vstup

AaBAba

výstup

5

V tomto príklade vystupujú dva druhy syra, druh *a* a druh *b*. Prvý dopyt po syre *a* je možné vyriešiť iba odkotúlaním syra z jediného iného domu naľavo od neho, čo vyžaduje kotúľanie o jeden meter.

Jediný dopyt po syre *b* vyžaduje 2 metre kotúľania a posledný dopyt po syre typu *a* vyžaduje 2 metre kotúľania, čo je dokopy 5 metrov.

vstup

AAaAaAAAAAaAaAAaAa

výstup

16

Všimnite si, že nie je potrebné, aby každý výrobca niekomu svoj syr predal, všetci konzumenti však musia byť uspokojení.

Odovzdávanie riešení

Zo stránky <http://oi.sk/> stiahnite ZIP archív obsahujúci 5 testovacích vstupov, nazvaných 1.txt až 5.txt.

Vyrobte k čo najviac vstupom správne výstupy a uložte ich do súborov sol1.txt až sol5.txt.

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

Za každý správny výstupný súbor získate 2 body.

B-I-2 Eliminácia

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.

Tento rok sa opäť koná Každoročná Súťaž v Pečení. Najlepší súťažiaci z celého sveta napiekli rozmanité torty, koláče a zákusky všetkých chutí, farieb a tvarov. Ostáva vybrať víťaza súťaže, ktorý vyhrá najmodernejší kuchynský robot a all-inclusive dovolenku v Neapole.

Víťazný koláč tento rok vyberú porotcovia Andráš a Sandra, ktorí majú úplne opačné názory na to, ktorý koláč je lepší ako iný. Andráš má rád veľké čokoládové torty, najlepšie ak sú posypané orechmi, a horká chuť je znakom pravého umenia. Sandra má najradšej maličkú ovocnú kapkejku, čo najsladšie, a na orechy je alergická.

Je jasné, že na víťazovi sa nezhodnú, tak sa dohodli, že víťaza vyberú nasledovným spôsobom. Poukladajú koláče na stôl do radu a budú sa striedať v ťahoch. Andráš začína. Keď je porotca na ťahu, príde k stolu a vylúči zo súťaže ľavú alebo pravú polovicu zostávajúcich koláčov. Posledný koláč, ktorý ostane, vyhráva.

Andraš chce, aby vyhral čo najlepší koláč podľa jeho názoru, Sandra zasa chce, aby vyhral čo najlepší koláč podľa jej názoru (čo je zároveň tým horší podľa Andrašá). Ktorý koláč vyhrá?

Súťažná úloha

Máme zadaný zoznam celých čísel – hodnotenia koláčov podľa Andrašá. Počet čísel je mocnina dvojky. Dvaja hráči Andráš a Sandra striedavo zahodia prvú alebo druhú polovicu zostávajúcich čísel, až kým neostane len jedno číslo. Andrášovým cieľom je, aby na konci ostalo čo najväčšie číslo, Sandriným aby ostalo číslo čo najmenšie.

Nájdite víťazné číslo x , ktoré im ostane na konci hry.



Obaja hráči hrajú perfektne racionálne a navzájom to o sebe vedia. To znamená, že číslo x je jednoznačne určené. Iný pohľad na hodnotu x je taký, že Andráš vie hrať tak, aby bez ohľadu na to, ako hrá Sandra, vyhralo číslo *aspoň* x . A naopak, Sandra vie hrať tak, aby bez ohľadu na to, ako hrá Andráš, vyhralo číslo *najviac* x .

Formát vstupu a výstupu

V prvom riadku vstupu je číslo n – počet koláčov. Hodnota n je mocninou dvojky.
V druhom riadku je n celých čísel od 1 po 10^9 , hodnotenia koláčov podľa Andráša. Na výstup vypíšete jedno číslo, aké hodnotenie má koláč, ktorý ostane na konci ako víťazný.

Príklad

vstup	výstup
4 1 1000 3 3	3

Keby Andráš zahodil pravú polovicu, Sandra by tiež zahodila pravú polovicu, a ostal by koláč s hodnotou 1. To Andráš nechce, preto radšej zahodí ľavú polovicu. Potom už je jedno, čo spraví Sandra, ostane koláč s hodnotou 3.

vstup	výstup
8 1 8 5 2 7 4 1 2	5

Po prvom tahu ostanú koláče 1 8 5 2, po Sandrinom tahu ostanú 5 2, a Andráš na konci nechá koláč s hodnotou 5. (Rozmyslite si, ako by hra dopadla, ak by Andráš spravil iný prvý, resp. Sandra iný druhý tah.)

Odvzdávanie riešení

Zo stránky <http://oi.sk/> stiahnite ZIP archív obsahujúci 10 testovacích vstupov, nazvaných 01.txt až 10.txt. Vyroberte 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 bod.

Počty koláčov v jednotlivých vstupoch sú dané tabuľkou:

vstup	1.txt	2.txt	3.txt	4.txt	5.txt	6.txt	7.txt	8.txt	9.txt	10.txt
n	8	16	512	1024	4096	32768	131072	262144	524288	1048576



B-I-3 Predvolebné fotenie

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

V dedine Lenivník je postavený veľký tehlový múr skladajúci sa z n úsekov. Keďže sa však oň nik nestará, je v katastrofálnom stave a postupne sa rozpadá. Obyvateľom Lenivníka začalo prekážať, že múr nemá na všetkých miestach rovnakú výšku, čo sa rozhodli využiť vo svojej predvolebnej kampani kandidáti na poslancov do obecného zastupiteľstva.

Každý z nich sľubuje, že múr zrekonštruuje tak, aby mal všade rovnakú výšku. (Každý kandidát si zvolil vlastnú preferovanú výšku h_i , na ktorú chce múr dať prerobiť.) Prázdne sľuby sú však už v Lenivníku dávno prekuknuté, kandidáti vo voľbách sa preto musia preukázať skutkami. V rámci možností obyvateľov Lenivníka, pravdaže.

Každý kandidát sa preto išiel odfotiť pred múr. Vybral si niekoľko vedľa seba stojacich úsekov, ktoré sa budú na pozadí dobre vynímať a k jednému z nich postavil robotníka. Samozrejme, robotník musí stáť pri úseku, ktorý je potrebné zrekonštruovať, tento úsek teda nemôže mať týmto kandidátom sľúbenú výšku h_i . Skvelá fotka na predvolebný plagát a jasný dôkaz, že sa s rekonštrukciou už začalo.

Ako správni obyvatelia Lenivníka, sú aj kandidáti do zastupiteľstva nesmierne leniví a nechce sa im premýšľať, ku ktorej časti múru majú postaviť robotníka. Rozhodli sa preto obrátiť na vás, *sľubujúc* odmenu, o akej svet ešte nepočul.

Súťažná úloha

Dostanete popis múru, teda n čísel reprezentujúcich výšku jednotlivých úsekov (v poradí, v ktorom tvoria múr). Každý z q kandidátov vám následne zadá časť múru (interval úsekov), s ktorou sa chce odfotiť a výšku múru h_i , ktorú sľubuje vo svojej kampani. Vašou úlohou je pre každého kandidáta nájsť a vypísať číslo ľubovoľného úseku múru, ktorý bude na fotke a ktorého výška nie je h_i .

Formát vstupu a výstupu

V prvom riadku vstupu sú celé kladné čísla n a q označujúce dĺžku múru a počet kandidátov. V druhom riadku je n medzerou oddelených celých kladných čísel v_1, \dots, v_n popisujúcich výšky múru v každom z n úsekov.

Nasleduje q požiadavok. Každú popisuje jeden riadok obsahujúci tri celé čísla z_i , k_i a h_i , kde z_i označuje začiatkový úsek a k_i koncový úsek intervalu, pri ktorom sa chce i -ty kandidát odfotiť. Číslo h_i označuje želanú výšku múru tohto kandidáta. Pre každú požiadavku platí $1 \leq z_i \leq k_i \leq n$.

Pre i -tu požiadavku je číslo x správnou odpoveďou práve vtedy, ak platí $z_i \leq x \leq k_i$ a zároveň $v_x \neq h_i$.

Na výstup vypíšte pre každú požiadavku (v poradí, v akom sú na vstupe) jednu správnou odpoveď pre ňu. Ak pre niektorú požiadavku existuje viac správnych odpovedí, môžete vypísať ľubovoľnú z nich. Ak pre niektorú požiadavku neexistuje žiadna správna odpoveď, namiesto nej vypíšte číslo -1 .

Obmedzenia a hodnotenie

Pri riešení dbajte na to, aby vami navrhnutý algoritmus našiel správne riešenie v každom možnom prípade. Nespoliehajte sa preto na náhodu a nepredpokladajte o vstupných dátach nič, čo nie je explicitne spomenuté v zadaní.

Plných 10 bodov môžu získať riešenia, ktoré efektívne vyriešia ľubovoľný vstup, v ktorom platí $n, q \leq 10^6$.

Za riešenia, ktoré efektívne vyriešia ľubovoľný vstup s $n, q \leq 10^5$ môžete získať najviac 7 bodov.

Za riešenia, ktoré efektívne vyriešia ľubovoľný vstup s $n, q \leq 5000$ môžete získať najviac 5 bodov.

Hodnoty h_i aj hodnoty v_j sú kladné celé čísla. Nepredpokladajte o nich nič iné ako to, že sa zmestia do bežnej celočíselnej premennej.



Príklady

vstup

```
10 4
4 3 3 3 2 3 3 5 5 4
4 5 2
4 10 8
2 4 3
1 5 3
```

výstup

```
4
8
-1
1
```

Prvý kandidát sa chce odfoťiť s úsekmi od 4 po 5 a hľadá taký, ktorého výška nie je 2. Taký je v tomto intervale jedine úsek 4, prvá odpoveď teda musí byť 4.

Pre druhého kandidáta je správnu odpoveďou ľubovoľná hodnota od 4 po 10, keďže žiaden z týchto úsekov nemá výšku 8.

Tretí kandidát sa odfoťiť nevie, keďže všetky úseky v intervale 2 až 4 majú výšku 3.

Štvrtý kandidát môže postaviť robotníka buď pred úsek číslo 1 alebo pred úsek číslo 5.

B-I-4 Pohovor

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

Andrejovým veľkým snom bolo získať prácu programátora v nemenovanej firme, ktorej šesťpísmenný názov začína písmenom G. Napísal si životopis, poslal ho do firmy G a na jeho veľké prekvapenie mu poslali pozvánku na pohovor, na ktorom ho preskúšajú z algoritmického a analytického rozmýšľania. Poctivo sa teda pripravoval a v daný deň sa nervózne pripojil do videohovoru. Zamestnanec firmy G, ktorý viedol pohovor, bol milý, ale nestrácal čas a rovno Andrejovi predostrel nasledovnú úlohu.

Majme číslo n . Toto číslo vieme meniť pomocou nasledovných operácií: pripočítanie jednotky, odčítanie jednotky, vydelenie dvomi (ak je číslo párne). Zistite, koľko najmenej operácií potrebujeme, aby sme z čísla n dostali číslo 1.

Andrej si spomenul, že najhoršie na pohovore je ostať ticho a je lepšie začať s horším riešením, ktoré následne vylepší aj s pomocou zamestnanca. Napísal preto takýto jednoduchý rekurzívny kód, ktorý skúšal všetky možnosti.

Listing programu (Python)

```
def rekurzia(n):
    if n == 1:
        return 0
    # najmenší počet krokov ak odčítam 1
    pocet_krokov = 1 + rekurzia(n - 1)
    # najmenší počet krokov ak pripočítam 1
    pocet_krokov = min(pocet_krokov, 1 + rekurzia(n + 1))
    # najmenší počet krokov ak vydělím 2
    if n % 2 == 0:
        pocet_krokov = min(pocet_krokov, 1 + rekurzia(n // 2))
    return pocet_krokov
```

Samozrejme, keď Andrej toto riešenie vyskúšal spustiť, bežalo donekonečna, pretože prehľadávalo aj úplne nezmyselné cesty. Bol to však začiatok a zamestnanec z firmy G sa ho spýtal nasledovné otázky, ktorými mal Andrej svoje riešenie vylepšiť. Jemu sa to podarilo a do firmy G ho naozaj prijali. Podarí sa to aj vám?¹

¹Zatiaľ teda myslíme aspoň to vylepšenie algoritmu, na kariéru ešte máte čas :)



Súťažná úloha

a) (3 body) Zdá sa, že pričasté pripočítavanie a odpočítavanie jednotky je neefektívne. Dokážte, že pre každé n existuje optimálne riešenie, ktoré nikdy nepoužije bezprostredne po sebe dve operácie, z ktorých každá je $+1$ alebo -1 .

b) (4 body) Andrej upravil svoje riešenie na základe tvrdenia v podúlohe a) tak, aby nikdy nepoužilo za sebou dve a viac operácií $+1$ a -1 . Zaujíma ho, či mu táto úprava vôbec pomohla.

Predstavme si, že sme pre dané číslo n postupne (či už programom alebo ručne) našli úplne všetky postupnosti operácií, ktoré majú vlastnosť z podúlohy a) a ktoré vedú k tomu, že na ich konci prvýkrát dosiahneme hodnotu 1.

Ďalej si predstavme, že si pre každú takúto postupnosť operácií vypíšeme všetky čísla, ktoré sme v priebehu jej aplikovania na číslo n videli.

V tejto podúlohe je vaším cieľom odhadnúť, koľko **rôznych** čísel si dokopy vypíšeme.

Napríklad nech $n = 7$. Postupnosti operácií (nie nutne najkratšie), ktorými sa dostaneme zo 7 k 1 sú nasledovné: $(+1, /2, /2, /2)$, $(+1, /2, /2, -1)$, $(-1, /2, +1, /2, /2)$, $(-1, /2, +1, /2, -1)$ a $(-1, /2, -1, /2)$. Keď ich aplikujeme, zapíšeme si všetky čísla od 1 po 8 okrem 5. (Číslo 7 si tiež vypíšeme, vždy na začiatku postupu.) Dokopy sme teda vypísali 7 rôznych čísel.

V tejto podúlohe **nemusíte** vypočítať **presnú** hodnotu, stačí keď ju odhadnete na základe čísla n rovnako, ako by ste odhadovali časovú zložitosť.

c) (2 + 1 bod) Pre čo najviac spomedzi nasledovných hodnôt čísla n zistíte najmenší počet operácií potrebný na dosiahnutie hodnoty 1:

- 9 685
- 9 120 157
- 770 228 663 542
- 944 169 986 423 364 522

V tejto podúlohe očakávame, že si pomôžete nejakým vami naprogramovaným algoritmom. Tento program vložte do vami spísaného riešenia. Dva body za podúlohu c) dostanete na základe správnych odpovedí pre vyššie uvedené čísla. Zvyšný bod vám bude udelený na základe programu, ktorý ste použili pri riešení.

TRIDSIATY ÔSMY ROČNÍK OLYMPIÁDY V INFORMATIKE

Príprava úloh: Michal Anderle, Jano Hozza, Andrej Korman, Timea Szöllősová, Samuel Gurský, Michal Farnbauer

Recenzia: Michal Forišek

Slovenská komisia Olympiády v informatike

Vydal: NIVAM – Národný inštitút vzdelávania a mládeže, Bratislava 2022