

Korešpondenčný seminár z programovania

Leták zimnej časti XXIX. ročníka

Korešpondenčný seminár z programovania (KSP) je súťaž programátorov – stredoškolákov a mladších – pripravovaná skupinou študentov a doktorandov FMFI UK. Jej cieľom je zdokonaľiť žiakov v programovaní a v algoritmickom myslení.

Ak študuješ na strednej škole a vieš aspoň trochu programovať, neváhaj a zapoj sa do našej súťaže, má to množstvo výhod:

- Riešením súťažných úloh a štúdiom našich vzorových riešení sa môžeš naučiť mnoho nového. Získané poznatky a skúsenosti sa ti iste budú hodiť v iných súťažiach v programovaní (napríklad pri riešení Olympiády v informatike), počas vysokoškolského štúdia, či pri prijímacích pohovoroch do zamestnania. (Mnoho našich bývalých riešiteľov sa bez ťažkostí zamestnalo v špičkových IT spoločnostiach ako Google, Facebook, ESET, ...)
- Na riešenie úloh máš dosť času a môžeš ich riešiť doma bez toho, aby si niekam cestoval.
- Medzi zadaniami sa nachádzajú ľahšie aj ťažšie. Každý si môže vybrať tie, ktoré vie riešiť a ktoré považuje za zaujímavé.
- Pre najlepších riešiteľov organizujeme každoročne dve týždenné sústreďenia. Sústreďenie je jedinečnou príležitosťou ako spoznať nových priateľov s podobnými záujmami, naučiť sa čosi viac nielen o programovaní a zažiť kopec zábavy.

Ako KSP prebieha?

Počas školského roka prebehnú dve samostatné časti súťaže: zimná a letná. Každá časť sa skladá z dvoch kôl, každé kolo obsahuje desať súťažných úloh. Najlepších riešiteľov zimnej časti pozývame na jarné sústreďenie; najlepších riešiteľov letnej časti zase na to jesenné.

Súťažiť sa dá v troch kategóriách: **Z** (pre začínajúcich riešiteľov, obsahuje úlohy 1–5), **O** (skúsenejší riešitelia, úlohy 4–8) a **T** (špeciálna kategória pre náročných, úlohy 9 a 10). Každá kategória má svoju vlastnú výsledkovú listinu. Na sústreďenia pozývame riešiteľov na základe výsledkov v kategóriách Z a O.

Vaše riešenia úloh môžete odovzdávať na stránke <http://www.ksp.sk/eRiesenie>, kým neuplynie termín určený v zaniatiach kola. Po tom, čo riešenia opravíme, nájdete na tomto mieste aj naše komentáre k nim a počet bodov získaný za jednotlivé úlohy.

Ako má vyzerať riešenie a za čo dostanem body?

Vašou úlohou je vytvoriť program, ktorý rieši zadanú úlohu. V prvom rade sa snažte, aby bol korektný, t.j. aby dal pre každý vstup správnu odpoveď, v druhom rade aby bol čo najrýchlejší a mal čo najmenešie pamäťové nároky.

Riešenie úlohy pozostáva z programu a popisu použitého algoritmu. V zadaní vždy uvedieme, koľko bodov sa dá získať za program a koľko za popis; výsledný počet bodov za úlohu je súčtom týchto dvoch hodnotení.

Váš program hneď po odovzdaní automaticky otestujeme na viacerých vopred pripravených vstupoch. Body za neho vám pridáme podľa toho, na koľkých vstupoch dá správnu odpoveď v časovom limite. Len čo sa program dotestuje, dozvieme sa výsledok. Ak ste nezískali plný počet bodov, môžete program vylepšiť a odovzdať ho znova. Podrobnejšie informácie o odovzdávaní programov nájdete na našej webstránke.

Popis algoritmu by mal byť natoľko podrobný a zrozumiteľný, aby bolo možné podľa neho napísať program rovnako efektívny, ako ten váš. Ďalej vyžadujeme odhad časovej a pamäťovej zložitosti a zdôvodnenie (ak je to potrebné, aj dôkaz) správnosti algoritmu.

Ak vo svojom riešení používate zložitejšie dátové štruktúry (napríklad haldy, nie obyčajné pole), musíte popísať aj ich implementáciu. To platí aj v prípade, že ich váš programovací jazyk už obsahuje a vy ste ich neimplementovali. Ak si nie ste istí, či niečo môžete použiť bez popisu, radšej to popíšte, prípadne sa pýtajte vo fóre na stránke.

Popis odovzdávajte vo formáte PDF alebo ako plain text. Hodnotíme hlavne korektnosť algoritmu a v druhom rade jeho efektívnosť. Získaný počet bodov sa dozvieme, keď vaše riešenie po termíne odovzdania opravíme.

Zopár ukážkovo vyriešených starších úloh nájdete na stránke <http://www.ksp.sk/wiki/Seminar/Riesenie>. Na tomto mieste sa tiež môžete dočítať, čo je vlastne časová a pamäťová zložitosť (ak vám tieto pojmy veľa nehovoria).

Ktoré kategórie môžem riešiť?

Kategóriu Z môžu riešiť:

- tretiaci a štvrtáci¹, ak do začiatku príslušného polroka neboli na sústredení KSP,
- druháci a mladší, ak do začiatku príslušného polroka boli najviac na jednom sústredení KSP.

Kategórii O a T sa môžu zúčastniť všetci stredoškólcovia (a mladší) bez obmedzenia. Ak vám to pravidlá dovoľujú, môžete riešiť aj viacero kategórií naraz.

Čo je také špeciálne na kategórii T?

Úlohou tejto kategórie je celoročná príprava riešiteľov na medzinárodné súťaže. Jej víťaza čaká večná sláva a hodnotná vecná cena (zvyčajne je to knižka podľa vlastného výberu). Navyše tento rok **budú body z kategórie T mierne zohľadnené pri výbere reprezentácie na Medzinárodnú olympiádu v informatike**. (Tento výber má formu týždňového sústredenia, na ktoré sú pozvaní najlepší riešitelia celoštátneho kola Olympiády v informatike, kat. A.)

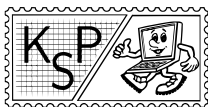
Na rozdiel od kategórií Z a O vám v kategórii T riešenie opravíme do niekoľkých dní od odovzdania. Ak ste nezískali plný počet bodov, môžete skúsiť na svojom riešení popracovať a potom ho odovzdať znovu; toto sa môže opakovať koľkokrát chcete (a stihniete). Navyše máte na úlohy kategórie T viac času: riešenia prvého aj druhého kola tejto kategórie môžete posielat až do uplynutia termínu druhého kola.

Registrácia

Pred odovzdaním elektronického riešenia je potrebné zaregistrovať sa na našej webstránke a vyplniť požadované kontaktné údaje. Odporúčame sa zaregistrovať pár dní pred dňom, kedy chcete odovzdať vaše riešenie (pre prípad, že by ste mali počas registrácie nejaké problémy).

Účasťou u KSP nám dávate súhlas spracovať a archivovať údaje, ktoré nám poskytnete pri registrácii, ako aj zverejniť vaše meno, školu, ročník a získané body vo výsledkovej listine.

¹ Za štvrtákov považujeme študentov, ktorí maturujú v tomto školskom roku; tretiaci sú tí, ktorí budú maturovať budúci školský rok; ostatné ročníky analogicky.



Úlohy 1. kola zimnej časti

Termín odoslania riešení tejto série je pondelok **24. októbra 2011.**

1. Zapáčiť sa Táni

kat. Z; 7 b za popis, 3 b za program

Ach, božská Táňa!

Táňa je pekná a šikovná, preto niet divu, že sa za ňou ohliada každý KSPák. Táňa si je ale vedomá svojich kvalít – nejde na rande so žiadnym chudákom! Vyjde si von iba s nadpriemerným chlapcom. Samozrejme, to aký je chlapec dobrý, je úplne jasné z pohľadu naň – chlapi sú priehľadní ako sklo. Toto všetko vedia aj tety bufetárky (Hanka a Dášenska), pretože majú dobré kontakty v KSP. A dobre vedia aj to, že keď chlapec niekam Táňu pozve, tak je to do bufetu na kofolu.

Dneska prišla Táňa na skúšku v úplne nových, neskutočne sexi šatách. Preto ju chce teraz úplne každý KSPák pozvať na rande. V bufete by sa radi pripravili na nával a potrebujú vedieť, koľko pozvaní Táňa prijme, aby objednali dosť kofoly.

Úloha

Pre každého KSPáka poznáme jeho faktor šmrncovnosti². Vašou úlohou je zistiť počet KSPákov, ktorí sú nadpriemerní. To znamená, že ich faktor šmrncovnosti je ostro väčší ako priemer. Tento priemer rátame iba z ľudí na vstupe.

Formát vstupu

Prvý riadok obsahuje jediné číslo n ($1 \leq n \leq 500\,000$) – počet KSPákov. V druhom riadku je n medzerami oddelených celých kladných čísel udávajúcich šmrnc jednotlivých KSPákov.

Kedže je KSP organizované matfyzákmi, môžete predpokladať, že celkový šmrnc semináru určite nepresiahne 10^9 .

Formát výstupu

Na výstup vypíšte jediné celé číslo – počet KSPákov, ktorých šmrncovnosť je nadpriemerná.

Príklad

vstup	výstup
5 3 7 5 61 19	1

Priemer čísel na vstupe je 19. Ako vidíme, Bob (61) je jediným KSPákom dostatočne šmrncovým na to, aby si s ňím Táňa vyšla. Míro (19) je priemerný, a preto oňho Tánička záujem nemá.

² kombinujúci v sebe inteligenciu, vzhľad a všetky kvality, ktoré Táňa hľadá

2. Záhradné rošády

kat. Z; 7 b za popis, 3 b za program

Slnko ešte stále pripeká a v záhradke zrejú plody. Mrkva a hrach sa síce skrývajú pred slnkom, ale kaleráby a patizóny si ho ešte užívajú. Ktovie, čo robí zelenina v noci? To nikto nevie, lebo v noci je tma. A ak v noci nie je tma, tak je spln a všetci sa skrývajú pred vlkodlakmi.

Ak to nikomu nepoviete, tak pravdou je, že v noci chodí Rasto tajne presádzať zeleninu. Konkrétne na mrkvovo-petržlenovej hriadke chce dosiahnuť svoju obľúbenú postupnosť mrkviem a petržlenov. Aby to nebolo nápadné, tak každú noc vymení len dve susedné zeleniny.

Úloha

Pre dané rozloženie mrkviem a petržlenov na hriadke vypočítajte minimálny počet nocí, ktoré potrebuje Rasto na to, aby dosiahol svoju obľúbenú mrkvovo-petržlenovú postupnosť.

Formát vstupu

Vstup sa skladá z troch riadkov. Prvý z nich obsahuje celé číslo n ($1 \leq n \leq 90\,000$) – počet kusov zeleniny na hriadke. V druhom riadku je zápis pôvodnej postupnosti zeleniny a v treťom riadku zase zápis Rastovej obľúbenej postupnosti.

Postupnosť mrkviem a petržlenov v hriadke zapíšeme ako reťazec znakov M a P dĺžky n . Pôvodná a výsledná postupnosť budú obsahovať rovnaký počet mrkviem (a teda aj rovnaký počet petržlenov).

Formát výstupu

Na výstup vypíšete jediné celé číslo – počet nocí, ktoré Rasto potrebuje k splneniu svojho úmyslu.

Príklad

vstup

```
5
PPMMM
MMPPM
```

výstup

```
4
```

Príklad presádzania:

PPMMM (2. <-> 3.)

PMPMM (1. <-> 2.)

MPMMM (3. <-> 4.)

MPMPM (2. <-> 3.)

MMPPM

3. Zbierka nepodarkov

kat. Z; 7 b za popis, 3 b za program

Peter varí špagety. Kúpil v obchode niekoľko balení rôznych značiek a ide spraviť veľa porcií. Každú značku charakterizuje nejaké kladné (ale nie nutne celé) číslo t : keď sa špagety tej značky varia t minút, budú akurát; keď viac ako t minút, budú rozvarené; a keď menej ako t minút, budú nedovarené. Rôzne značky môžu mať rôzne t , ale všetky špagety rovnakej značky majú rovnaké t .

Peter nevie, koľko je t , takže si dĺžku varenia musel tipnúť. Bohužiaľ si pre každú porciu tipol úplne zle a žiadna sa mu nepodarila akurát. Nahnevaný Peter sa rozhodol, že tieto značky si už nikdy nekúpi. Lenže obaly už zahodil a nevie ani, koľko rôznych značiek vlastne mal. Pamätá si len, že z každej značky skončila aspoň jedna porcia nedovarená a aspoň jedna rozvarená.

Úloha

O každej porcii viete, koľko minút ju Peter varil a ako skončila. Žiadne dve porcie nevaril rovnako dlho. Zistite, najmenej koľko značiek špagiet musel Peter mať.

Formát vstupu

Na prvom riadku sú dve kladné celé čísla n a r ($1 \leq n, r \leq 10\,000$) – počet nedovarených a rozvarených porcií.

Na druhom (tretom) riadku je n (r) kladných celých čísel – počet minút, ako dlho sa varili jednotlivé nedovarené (rozvarené) porcie. Žiadna porcia sa nevarila dlhšie ako $100\,000$ minút³.

Formát výstupu

Ak situácia na vstupe nemohla nastať, vypíšte -1 . Inak vypíšte riadok s jedným celým číslom – minimálnym počtom značiek špagiet.

Príklady

vstup

```
2 3
2 4
5 6 3
```

výstup

2

Jedna značka by nestačila, ale dve už hej. Napríklad z prvej značky spravil porcie 2 a 3, lenže mala $t = 2.5$, a z druhej značky spravil 4, 5 a 6, lenže mala $t = 4.5$. Preto sa 2 a 4 nedovarili a 3, 5 a 6 rozvarili.

vstup

```
1 1
5
4
```

výstup

-1

Podmienku, že z každej značky sa aspoň jedna porcia nedovarila a aspoň jedna rozvarila, nejde splniť.

vstup

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

výstup

1

Napríklad mohlo byť $t = 4$.

4. Zabalíme optimálne

kat. Z a O; 10 b za popis, 5 b za program

Tak sa nám pomaly skončilo leto, vás to nahnalo naspäť do školy a KSPákov zase rozbehnúť váš obľúbený seminár. Oddýchnutí a plní dojmov sa teda znova stretli a obohacovali všeobecné vzdelanie ostatných o svoje zistenia. Taký Mišo si raz vyšiel do prírody a zistil, že slimák záhradný (*Helix pomatia*) má dokonale prepracovaný školský systém. Podporuje totiž výučbu mimo školského areálu, čo zabezpečuje, že mozog detí neupadá do stereotypu.

V deň, keď Mišo pozoroval slimačiu školu, pani učiteľka práve chystala deti na matematický výlet. Vybrala zo školskej knižnice krabicu s číslami a nabalila ich deťom do ulít. Ulitky sú však ešte malé, a preto sa tam zmestí len jedno číslo. Našťastie sme v 21. storočí, technika pokročila, teda čísla sa vedia skomprimovať.

³veď desať týždňov už predsa musí stačiť

Do každého čísla sa vie zabaliť ľubovoľné **jedno menšie** číslo. Do neho sa vie zasa zabaliť ešte menšie číslo, atď.⁴ Napríklad môžeme zabaliť dvojku do trojky a tú do päťky; takto dostaneme tri čísla v jednom balíčku. Čísla 1, 1, 2 sa nám však do jediného balíčka zabaliť nepodari.

Deti sú ale hyperaktívne a tak sa triviálne prerozdelenie týchto balíčkov čísel stáva vcelku namáhavou činnosťou. Mišo by chcel pani učiteľke pomôcť, ale keď už je to taká školácka úloha, prečo by sme to nedali vyriešiť vám?

Úloha

Zistite, najmenej koľko balíčkov potrebujeme na zabalenie všetkých čísel.

Počet čísel zabalených v jednom balíčku nazvime *veľkosťou* tohto balíčka. V druhom rade chceme minimalizovať veľkosť najväčšieho balíčka⁵ (stále za podmienky minimálneho celkového počtu balíčkov).

Formát vstupu

Prvý riadok vstupu obsahuje jedno celé číslo n ($1 \leq n \leq 100\,000$) – počet čísel, ktoré chce pani učiteľka zabaliť.

Tieto čísla sú zadané v druhom riadku, každé z nich je celé a leží v intervale $[1, 10^9]$ (vrátane). Navyše, vo väčšine vstupov tieto čísla nepresiahnu 100 000.

Formát výstupu

Na prvý riadok výstupu vypíšete najmenší počet balíčkov k , ktorý stačí na zabalenie všetkých čísel. Na druhý riadok vypíšete minimálnu veľkosť najväčšieho balíčka, ktorú vieme dosiahnuť pri použití k balíčkov.

Príklad

vstup	výstup
11 2 5 7 6 5 8 9 10 4 8 3	2 6

Do jedného balíčka sa nám všetky čísla zjavne nevojdú. Rozdelíme ich preto na dve časti: (2, 3, 4, 5, 6, 8) a (5, 7, 8, 9, 10). Najväčšia časť má veľkosť 6 (čo je určite optimálne, keďže čísel je dokopy 11).

5. O sedlákov

kat. Z a O; 2 b za popis, 13 b za program

Sedlák, sedlák, nasadil stromy, kvety a k tomu iné veci stupidne... Ale to je iná pesnička.

V Dedinskej Novej Vsi žije sedlák Fero. Má malý domček, veľkú záhradku a rád experimentuje s krížením rastlín. Minule sa mu podarilo skrížiť banánovník s pomarančovníkom. Tento banáno-pomarančovník má však nezvyčajné vlastnosti. Rýchlo sa množí a ovocie rodí nasledujúcim spôsobom: Naraz na ňom dozreje veľa banánov aj pomarančov. Strom sa potom otrasie a všetko zhodí naraz dole.

Keďže Fero má už stromov plnú záhradu a banány ani pomaranče už nemá kam dávať, rozhodol sa ich predávať vo veľkom. Po celej záhradke porozkladal debničky, do ktorých padajú banány a pomaranče. Jediným problémom je, že banány a pomaranče sú zmiešané a potreboval

⁴určite poznáte princíp matriošiek

⁵Občas sa na takom slimačom výlete stane nešťastie – poznáte to, kde-tu chodia bicykle a autá – preto chceme minimalizovať straty na školskom majetku.

by ich od seba nejak oddeliť. Sám to už nestíha a potreboval by to nejak automatizovať. A to už je vaša úloha.

Úloha

Na vstupe sú čiernobiely obrázky ovocia v debničke. Pre každý obrázok treba rozhodnúť, či je to banán, alebo pomaranč.

Formát vstupu

Prvý riadok obsahuje číslo n ($1 \leq n \leq 10$) – počet kusov ovocia v debničke. Ďalej nasleduje n čiernobielych obrázkov ovocia.

Každý obrázok je v nasledujúcom formáte: V prvom riadku budú dve čísla s a r ($10 \leq s, r \leq 1000$). V každom z nasledujúcich r riadkov bude s čísel oddelených medzerami – 1 reprezentuje čierny pixel a 0 zasa biely. Môžete predpokladať, že objekt v obrázku bude mať veľkosť aspoň 10×10 pixlov a navyše môžete predpokladať, že každý čierny pixel patrí banánu resp. pomaranču a jeho okraj je súvislý. V prvých 6 vstupoch bude banán vyzeráť podobne ako ten z príkladu.

Formát výstupu

Na výstup vypíšete n riadkov zodpovedajúcich jednotlivým obrázkom ovocia (v poradí ako na vstupe). Podľa druhu ovocia na obrázku vypíšete **banan** alebo **pomaranč**.

Príklady

vstup

```
1
17 20
0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0
0 1 1 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0
0 0 1 1 0 1 1 0 0 0 0 1 1 1 1 0 0 0
0 0 0 1 1 0 0 1 1 1 1 0 0 0 0 1 1 1
0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 1 1
0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1
0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0
```

výstup

```
banan
```

vstup

výstup

```

1
10 10
0 0 0 0 1 1 0 0 0 0
0 0 1 1 1 1 1 1 0 0
0 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 0
0 0 1 1 1 1 1 1 0 0
0 0 0 0 1 1 0 0 0 0

```

pomaranc

6. Obe(d/t)ovanie kravičiek

kat. O; 12 b za popis, 8 b za program

Za dedinou je hora, v hore jaskyňa a v jaskyni drak. Aby im drak nevyžieral panny (ktorých na dedine zrovna nazvyš nemajú), dohodli sa s ním dedinčania, že ho budú prikrmovať kravami z obecného stáda. Obecné stádo má momentálne n kráv. Podľa dohody s richtárom si drak každý deň môže objednať jedno z m dohodnutých menu (napr. 200 kíl sviečkovej na smotane). Na i -te menu padne k_i kráv.

Drak má ale problém: skôr či neskôr by vyžral všetky kravy v dedine, a potom čo? Kravy sa bohužiaľ samé od seba nenamnožia – dedina nemá udelenú výsadu vlastníť býka. Jediný spôsob, ako sa dostať ku viac krávam, je poslať richtára, nech kráľa o nejaké poprosí.

Kráľ funguje veľmi jednoducho a predvídateľne. Má s svojich obľúbených čísel $t_1, \dots, t_s$. Ak aktuálny počet kráv v dedine nie je kráľovo obľúbené číslo, kráľ richtára vyhodí a nič mu nedá. V opačnom prípade je kráľ počtom kráv potešený a richtárovi daruje d ďalších.

Drak môže vyštváť richtára za kráľom kedy len chce, aj postupne viackrát medzi dvoma svojimi jedlami.

Vypočítajte, koľko najviac kráv môže drak za svoj (teoreticky nekonečne dlhý) život zožrať.

Formát vstupu

V prvom riadku vstupu sú štyri celé čísla: aktuálny počet kráv n ($1 \leq n \leq 40\,000$), počet dračích menu m ($1 \leq m \leq 50$), počet kráľových obľúbených čísel s ($1 \leq s \leq 50$) a počet kráv, ktoré pridáva d ($1 \leq d \leq 100$).

V druhom riadku je postupnosť m celých čísel $k_1, \dots, k_m$ ($1 \leq k_i \leq n$) oddelených medzerami – počty kráv, ktoré môže drak v rámci rôznych menu zožrať.

V treťom riadku je postupnosť s celých čísel $t_1, \dots, t_s$ ($1 \leq t_i \leq n$) oddelených medzerami – kráľove obľúbené čísla.

Formát výstupu

Vypíšte jeden riadok a v ňom jedno celé číslo: najväčší počet kráv, ktoré môže drak dokopy zožrať. Ak existuje stratégia, ktorá drakovi umožní žrať do nekonečna, vypíšte namiesto toho jeden riadok so slovom „nekonečno“.

Príklady

vstup

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

výstup

12

Drak môže postupovať nasledovne: Zožerie 3 kravy a ešte 3 kravy. V dedine ostali 4 kravy, vyženie teda richtára za kráľom a ten dovedie piatu kravu. Drak zožerie ešte tri kravy, znovu pošle richtára, a nakoniec zožerie posledné tri kravy v dedine.

vstup

```
10 1 1 4
1
7
```

výstup

nekonecno

Drak postupne žerie kravy po jednej. Vždy, keď počet kráv klesne na 7, pošle richtára po štyri ďalšie.

7. Obrovská šachovnica

kat. O; 12 b za popis, 8 b za program

Vonku je strašné počasie, preto sa Pat a Mat hrajú na veľkom štvorčekovom papieri. Niektoré jeho políčka sú zafarbené, niektoré nie. Keďže si nevedia spomenúť, akú hru to na ňom predtým hrali, povedali si, že si teraz zahrajú šach.

Šachovnica je štvorec veľkosti $n \times n$ políček, pričom z každých dvoch políček susediacich stranou je **práve jedno** zafarbené. Najmenšie šachovnice teda vyzerajú takto:

```
0   1   10   01   010   101
      01   10   101   010   ...
          010   101
```

ALADIN sľubuje rovnako škaredé počasie ešte počas niekoľkých nasledujúcich dní, preto si chcú Pat a Mat zahrať na čo najväčšej šachovnici nachádzajúcej sa na papieri. Zistíte jej veľkosť.

Formát vstupu

V prvom riadku vstupu sú dve celé čísla r a s ($1 \leq r, s \leq 6000$), udávajúce počet riadkov a stĺpcov papiera. Ďalej nasleduje r riadkov a v každom je s znakov vyjadrujúcich, či je políčko zafarbené (1), alebo nie (0).

Formát výstupu

Na výstup vypíšte jediné číslo – dĺžku strany najväčšej šachovnice, ktorú možno nájsť na štvorčekovom papieri.

Príklady

vstup

```
4 5
01010
10101
01010
10101
```

výstup

4

Na papieri sú dokonca dve šachovnice so stranou 4.

vstup

výstup

```
4 5
10101
01110
10101
01010
```

2

Kvôli zafarbenému políčku v druhom riadku a treťom stĺpci neexistuje šachovnica so stranou 3.

8. Ostreľovanie

kat. O; 15 b za popis, 10 b za program

Zemčovi sa v poslednom čase akosi lepí smola na päty. Najprv sa omylom prihlásil do Ozbroyených síl Slovenskej republiky. Jeho následný e-mail, v ktorom vysvetlil svoju chybu, však ktosi v generálnom štábe vytlačil, skartoval a vymazal z počítača. A tak Zemčo musel nastúpiť na svoj výcvik.

Ale ani tam sa svojej smoly nezbavil. Pri tréningu strelby vyfasoval staručký AK-47, ktorý držal pokope iba silou vôle (a tiež vďaka hrubej vrstve lepiacej pásky). Pretože sa im práve minuli bežné okružle terče, pre Zemča zostal už len taký krivý (ale zato nekonečne veľký).

Pomôžte Zemčovi spočítať, ako sa mu darilo pri jeho tréningu.

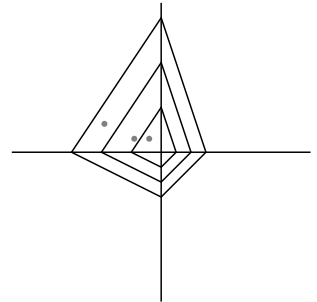
Úloha

Terč si môžeme predstaviť ako nekonečne veľa do seba vložených konvexných n -uholníkov v rovine.

Najvnútornejší má súradnice vrcholov $[x_1, y_1], [x_2, y_2], \dots, [x_n, y_n]$. Navyše sa v jeho vnútri určite nachádza bod $[0, 0]$. Za zásah do vnútra tejto oblasti dostane Zemčo 0 trestných bodov. Ďalšia oblasť má súradnice vrcholov $[2x_1, 2y_1], [2x_2, 2y_2], \dots, [2x_n, 2y_n]$. Za zásah do nej dostane Zemčo 1 trestný bod. Takto je ďalej navrstvených nekonečne veľa ďalších oblastí.

Formálne: ak sa zásah $[x, y]$ nachádza vo vnútri n -uholníka s vrcholmi $[(k+1)x_1, (k+1)y_1], [(k+1)x_2, (k+1)y_2], \dots, [(k+1)x_n, (k+1)y_n]$, ale už nepatrí do n -uholníka s vrcholmi $[kx_1, ky_1], [kx_2, ky_2], \dots, [kx_n, ky_n]$, tak Zemčo získa k trestných bodov.

Príklad situácie môžeme vidieť na obrázku. Zemčo získa dokopy $0 + 1 + 2 = 3$ trestné body.



Formát vstupu

V prvom riadku vstupu je číslo n ($4 \leq n \leq 100\,000$). Nasleduje n riadkov, v ktorých sú popísané vrcholy vnútornej oblasti, zoradené v smere hodinových ručičiek. Sú to dvojice celých čísel x_i, y_i ($-2\,000\,000\,000 \leq x_i, y_i \leq 2\,000\,000\,000$).

Potom je na vstupe riadok s číslom m – počet zásahov ($1 \leq m \leq 1\,000\,000$). Nasleduje m riadkov, ktoré popisujú súradnice zásahov. Sú to dvojice celých čísel x_i^z, y_i^z ($-2\,000\,000\,000 \leq x_i^z, y_i^z \leq 2\,000\,000\,000$). Môžete predpokladať, že žiadny zásah sa nebude nachádzať presne na okraji oblasti.

Formát výstupu

Na výstup vypíšte jedno celé číslo – počet trestných bodov, ktoré Zemčo nazbieral.

Príklad

vstup

```
4
1 0
0 -1
-2 0
0 3
3
-1 1
-2 1
-4 2
```

výstup

3

Zodpovedá obrázku zo zadania.

9. Tisíce podpostupností

kat. T; 10 b za popis, 15 b za program

Hovoríme, že postupnosť je *skáčuca*, ak je neprázdna a platí, že sa v nej žiadne dva bezprostredne po sebe nasledujúce prvky nerovnajú. Napríklad postupnosť $(1, 3, 1, 2)$ je skáčuca, ale $(2, 1, 1, 3)$ skáčuca nie je.

Podpostupnosť vznikne z postupnosti tak, že niekoľko členov (možno aj 0 a možno všetky) vyškrtne. Každá postupnosť dĺžky n má teda presne 2^n podpostupností. Vašou úlohou bude pre danú postupnosť spočítať, koľko z jej podpostupností je skáčucich.

Formát vstupu

V prvom riadku vstupu je celé číslo n , udávajúce dĺžku danej postupnosti. V druhom riadku je n medzerami oddelených celých čísel $a_1, \dots, a_n$ – prvky danej postupnosti.

Platí $1 \leq n \leq 500\,000$. Všetky a_i sú z rozsahu od 1 do 10^9 .

V niektorých vstupoch je n do 5 000.

Vo veľkej väčšine vstupov sú hodnoty a_i z rozsahu od 1 do 500 000.

Formát výstupu

Označme x hľadaný počet skáčucich podpostupností. Aby sa nebolo treba trápiť s veľkými číslami, stačí nám, keď spočítate hodnotu $y = (x \bmod 1\,000\,000\,007)$. Vypíšte jeden riadok a v ňom číslo y .

Príklad

vstup

```
4
1 2 1 1
```

výstup

9

Ide o nasledujúce podpostupnosti:

Tri majú tvar (1) , jedna (2) , jedna $(1, 2)$, dve $(2, 1)$ a dve $(1, 2, 1)$.

10. Totálna fúzia

kat. T; 15 b za popis, 10 b za program

V krajine menom Absurdistan čulo prekvitá cestovný ruch. Napríklad tam majú až ℓ leteckých spoločností a každá z nich prevádzkuje niekoľko letov po krajine. Dokonca platí, že z každého mesta krajiny sa do každého iného dá dostať len pomocou (jedného alebo postupne viacerých) letov.

Oficiálne má každá letecká spoločnosť iného vlastníka. V skutočnosti sú však všetky letecké spoločnosti samozrejme písané na ℓ manželiek veľkého vezíra Hassana.

V rámci šetrenia sa Hassan nedávno rozhodol spraviť dve úsporné opatrenia. Prvým bolo, že zrušil Protimonopolný úrad. A potom mu už nič nebránilo spraviť druhé úsporné opatrenie – ďalej prevádzkovať bude len jednu z ℓ doteraz existujúcich spoločností. Niektoré lety ostatných aerolíniek začne odteraz prevádzkovať zostávajúca spoločnosť, ostatné lety zruší.

Aby sa ľud v krajine nevzbúril, musí Hassan zabezpečiť, že aj po jeho zásahu sa bude v krajine dať medzi každými dvoma mestami dostať pomocou letov. Hassan chce mať pri zmene aerolíniek čo najmenšie náklady. Nie že by ho trápila kríza, má peňazi na rozhadzovanie, ale malá kopa pýta viac.

Formát vstupu

V prvom riadku vstupu sú tri čísla oddelené medzerou: počet aerolíniek ℓ ($1 \leq \ell \leq 2000$), počet miest n ($2 \leq n \leq 2000$) a celkový počet letov f ($1 \leq f \leq 200\,000$). Mestá majú čísla 1 až n , aerolinky majú čísla 1 až ℓ .

Nasledujúcich f riadkov popisuje jednotlivé lety. Pre každý let i sú dané štyri čísla: a_i b_i (čísla miest, medzi ktorými tam a späť premáva), p_i (číсло aerolinky, ktorá ho prevádzkuje) a c_i (jednorazové náklady, ktoré musí Hassan vynaložiť, keď chce tento let začať prevádzkovať inou spoločnosťou). Platí $1 \leq a_i, b_i \leq n$, $a_i \neq b_i$, $1 \leq p_i \leq \ell$, $1 \leq c_i \leq 10^6$.

Formát výstupu

Na výstup vypíšte do jediného riadku jedno číslo: najmenšie celkové náklady potrebné na uskutočnenie Hassanovho plánu.

Príklad

vstup

```
3 4 4
2 3 1 6000
4 3 2 7000
1 2 2 3000
1 3 3 5000
```

výstup

5000

Optimálne pre Hassana je ponechať si spoločnosť 2. Tá prevádzkuje lety 1-2 a 3-4. Aby sa dalo dostať odváďať všade, je potrebné pridať ešte jeden zo zvyšných dvoch letov. Lacnejšie vyjde pridať let 1-3, čo nás stojí 5000.