

Korespondenčný seminár z programovania

Leták letnej časti XXIX. ročníka

Korespondenčný seminár z programovania (KSP) je súťaž programátorov – stredoškôľakov 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 najmenšie 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, dozviete 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 haldu, 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 samostatne (nie ako komentár v kóde) vo formáte PDF alebo ako plain text. Hodnotíme hlavne korektnosť algoritmu a v druhom rade jeho efektívnosť. Získaný počet bodov sa dozviete, keď vaše riešenie po termíne odovzdania opravíme.

Zopár ukázkovo 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 zložitnosť (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 medzinárodnú súťaž. 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.)

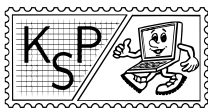
Na rozdiel od kategórií Z a O vám v kategórii T riešenie opravíme do niekoľkých dní od odovzdania. Ak ste nezískali plný počet bodov, môžete skúsiť na svojom riešení popracovať a potom ho odovzdať znovu; toto sa môže opakovať kôľkokrát chcete (a stihnete). Navyše máte na úlohy kategórie T viac času: 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 v 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.

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



Úlohy 1. kola letnej časti

Termín odoslania riešení tejto série je pondelok 12. marca 2012.

1. Zavádzame plánovanie!

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

Vedúci sa rozhodli, že v KSP zavedú plánované hospodárstvo. A tak hneď zvolali stretnutie, aby naplánovali svoju činnosť na päť rokov dopredu. Ale päť rokov, to znamená desať sústredení! Musia teda vymyslieť, kde sa sústredenia budú odohrávať. Usáma navrhol jednoducho vybrať desať náhodných slovenských obcí. Ale ostatní hneď namietali a prišli s dlhým zoznamom obcí, ktoré sa im nepáčia. Škoda, že nikomu nenapadlo vytlačiť si abecedný zoznam miest a obcí Slovenska a jednoducho tie zlé vyškrtiť. Namiesto toho teraz majú iba haldu papierov popísanú miestami, kde sústredenie nemôže byť. Pomôžete im zodpovedať otázku, kde vlastne sústredenie *môže* byť?

Formát vstupu

V prvom riadku vstupu sú dve čísla, n a k . Číslo n , kde $10 < n \leq 10^6$, je počet slovenských obcí. Obce sú očíslované od 1 po n . Číslo k udáva počet obcí, kde sústredenie nechceme, a platí $1 \leq k \leq n - 10$. Na druhom riadku vstupu je k rôznych čísel a_1, \dots, a_k – poradové čísla týchto obcí.

Formát výstupu

Na výstup vypíšte ľubovoľných 10 rôznych medzerou oddelených čísel, ktoré sú z rozsahu 1 až n , ale nenachádzajú sa medzi číslami a_1, \dots, a_k .

Príklad

vstup

```
100 4
3 20 47 5
```

výstup

```
1 2 4 6 7 8 9 10 53 42
```

2. Zabudnutá hra

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

Keď sa Žaba minule vrátil späť na svoju rodnú hru – do Lučenca, nestihol sa diviť, koľko vecí sa tam zmenilo. Napríklad im pribudol druhý semafor, ktorý už dokonca aj svieti. V rámci nostalgie si doma našiel krabicu so svojimi detskými hračkami. A na úplnom spodku našiel svoju obľúbenú hru, na ktorú už skoro zabudol.

Hra sa volá (doplňte názov možno aj link) a pozostáva z mriežky $n \times n$, v ktorej sú uložené pohyblivé čísla 1 až $n^2 - 1$ a jedno prázdne políčko. V každom ťahu sa môže posunúť jedno číslo, ktoré susedí s prázdnyim políčkom, na toto miesto, čím sa vytvorí nové prázdne políčko. Cieľom hry je usporiadať čísla do správneho poradia.

Žaba však okrem samotnej hry našiel aj papier, na ktorý si vždy zapísal začiatočné rozostavenie mriežky a postupnosť ťahov, ktorú vykonal. Nezapísal si však výsledné rozostavenie, a keďže nemá čas, lebo sa ponáhľa za kamarátkou Peťou², poprosil vás o pomoc. . .

Formát vstupu

Na vstupe dostanete v prvom riadku číslo n ($2 \leq n \leq 1\,000$) – rozmer mriežky. Na každom z nasledujúcich n riadkov je n čísel predstavujúcich začiatočné rozloženie mriežky. Špeciálne 0 predstavuje prázdne políčko. Môžete predpokladať, že vstup je správny a teda každé číslo od 0 po $n^2 - 1$ sa tam vyskytne práve raz.

Po popise mriežky nasleduje číslo k ($0 \leq k \leq 1\,000\,000$) a za ním k čísel predstavujúcich jednotlivé ťahy. Ťahy sa vykonávajú v poradí, v akom sú na vstupe, a každý ťah je reprezentovaný číslom od 1 po $n^2 - 1$, ktoré znamená, že chceme pohnúť týmto číslom. Ak sa vedľa neho nachádza prázdne políčko, tak sa naň posunie, inak zostane stáť na mieste.

Formát výstupu

Na výstup vypíšte n riadkov a na každom n čísel, predstavujúcich výslednú mriežku. Opäť 0 reprezentuje prázdne políčko.

Príklad

vstup

3
1 5 2
4 0 6
8 7 3
3
7
2
3

výstup

1 5 2
4 7 6
8 3 0

Najskôr sa číslo 7 posunulo o jedno políčko vyššie. Keďže vedľa čísla 2 nebolo prázdne políčko, tak zostalo na mieste. Nakoniec sa číslo 3 posunulo doľava na prázdne políčko.

3. Zaplav to!

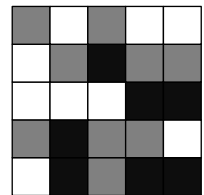
kat. Z; 12 b za popis, 0 b za program

Väčšina vedúcich KSP už nejaký ten piatok vlastní smartfón. A čo si budeme navrávať, jedno z hlavných použití smartfónov sú hry. Stálicou na tomto nebi je hra *Drag Toilet Paper*, tú majú nainštalovanú snáď všetci. Ale potom sa už preferencie líšia. Pár prívržencov má *Angry birds*, ale nezabúdajme ani na perly ako *Abduction* alebo *Air Control Lite*. No a Usameca môžete občas vidieť, ako kliká do hry *Flood-It!*. A s touto hrou súvisí aj táto naša úloha.³

Hrací plán v hre *Flood-It!* má tvar veľkého štvorca rozdeleného na $n \times n$ malých štvorcov. Každý z týchto štvorcov na začiatku dostane jednu náhodnú farbu spomedzi f možných.

Samozrejme, z času na čas sa takto stane, že dva susedné štvorce dostanú tú istú farbu. A občas takto vzniknú aj tri či štyri štvorce pokope. Tým na hracom pláne vznikajú veľké jednofarebné oblasti. A vtedy sa Usamec poteší, lebo čím väčšia jednofarebná oblasť na pláne vznikne, tým sa potom príslušný level dá rýchlejšie vyhrať.

(Na obrázku vpravo je príklad pre $n = 5$ a $f = 3$. Vidíme 15 jednofarebných oblastí, najväčšiu z nich tvoria 4 štvorce.)



²drb???

³Nebojte sa, nepotrebuje ju vedieť hrať. Ale ak si ju zahrať chcete, nebránime vám v tom :-). A netreba na to ani mobil, existuje aj veľa jej klonov na webe.

Úloha

Tvojou úlohou je zistiť, ako závisí od čísel n a f veľkosť najväčšej jednofarebnej oblasti na hracom pláne. Toto je samozrejme otázka veľmi nejednoznačná, lebo odpoveď na ňu závisí od toho, ako sa nám pošťastí pri náhodnom generovaní. A presne popísať túto závislosť je zase otázka veľmi ťažká. Takže čo to vlastne od teba chceme?

Napiš program, ktorý bude vedieť načítať n a f , vygenerovať hrací plán a vypočítať veľkosť najväčšej jednofarebnej oblasti na ňom. Pomocou tohto programu potom odmeraj, ako hľadaná veľkosť závisí od premenných n a f . Pokús sa aj odhadnúť, čo by sa dialo pre väčšie n ako tvoj program zvláda spracovať.

A čo nám máš poslať? Svoj program a výsledky, ku ktorým dospeješ s jeho pomocou. Obzvlášť oceníme, ak budú výsledky spracované do peknej prehľadnej tabuľky a tiež ako vhodný graf. Pre počet farieb f stačí uvažovať hodnoty od 2 do 8. Veľkosti strán hracieho plánu vyskúšaj rôzne, vrátane čo najväčších.

Príklad

To, čo nám pošleš, by mohlo vyzeráť nejak takto. Ale poriadnejšie spravené :-).

„Ahojte KSPáci. Napísala som si ten program a potom som ho 10000-krát spustila pre $n = 30$ a $f = 6$. Dostala som celú kopu rôznych číseliek: niekedy mala najväčšia oblasť veľkosť len 5, niekedy dokonca až 20. Najčastejšie to bolo 7, 8, alebo 9. Tak nejak. Priemer vyšiel okolo 8.6. Pre väčšie n som to už neskúšala, ale tipujem, že by mohlo platiť, že ten priemer bude vždy vychádzať tak rádovo n/π .“

4. Zamyslený alchymista

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

Kde bolo, tam bolo, alebo možno ani nebolo, za siedmimi deriváciami a siedmimi integrálmi v n -tej dimenzii žil (a možno ešte žije) jeden dávnozabudnutý alchymista. Keďže práca s chemickými prvkami nie je vždy bezpečná a on je neskutočný workoholik, jeho pokusy ho zanesli až sem, za hranice sveta bežných ľudí.

Raz si tak nad svojimi skúmvkami zaspomínal, ako začínal. Bolo len pár prvkov, ktorými ho sama príroda obdarila, a on sa rozhodol vytvoriť niečo nové. Vždy veril v mágiu čísla dva – veď sú dva póly, plus a mínus, dva organizmy, čo vedeli splodiť tretí. A to chcel – niečo tretie. Preto sa rozhodol držať zásady, že všetko, čo vytvorí, bude vždy vznikáť z dvoch prvkov. A tak začala jeho neúnavná robota. Skúšal všetky možné kombinácie – niektoré boli úspešné a objavil novotu, pár z nich mu zas dalo iba to, čo už mal, a ostatné vôbec nezreagovali.

A tak sa venoval prvkom tu a tam, vôbec nie systematicky. Keď ho tu zrazu zaujala myšlienka pohľadu z druhej strany. Keby poznal všetky pravidlá (spôsoby vytvorenia prvkov; reakcie), aké by bolo najkratšie odvodenie toho či tamtoho prvku?

Úloha

Začínate s n rôznymi prvkami a máte m pravidiel. Každého prvku máte k dispozícii neobmedzené množstvo. Pravidlá vyzerajú tak, že vždy z práve dvoch prvkov (nie nutne rôznych) viete vyrobiť jeden ďalší prvok. Ten však nemusí byť pre vás nový (tzn. taký už vyrobený máte).

Alchymista chce vedieť, ako rýchlo sa dá ktorý prvok vyrobiť. Niektoré sa nebudú dať vyrobiť ihneď, lebo si najprv budete musieť vyrobiť prvky, z ktorých sa vyrábajú. Vašou úlohou je vypísať, na koľko najmenej krokov viete vyrobiť i -ty prvok s tým, že v jednom kroku viete použiť aj viac pravidiel naraz, ak už máte odvodené prvky, ktoré v nich používate.

Formát vstupu

Prvý riadok vstupu obsahuje čísla k, n, m ($1 \leq k, n, m \leq 200\,000$). Prvky sú označené číslami od 1 do k . V druhom riadku je vypísaných n počiatočných prvkov. Ďalších m riadkov obsahuje pravidlá v tvare $a b c$, čo znamená, že ak skombinujete a a b , vyrobíte c (môže nastať aj prípad $a = b$).

Formát výstupu

Vypíšte po poradí počet krokov odvodenia každého prvku – pre každý prvok jeden riadok. Môžete predpokladať, že všetky prvky sa dajú odvodiť.

Príklad

vstup	výstup
6 2 5	0
1 2	0
1 2 3	1
1 1 4	1
2 4 5	2
4 4 2	3
3 5 6	

Prvky 1, 2 máme na začiatku. Prvok 3 vyrobíme pomocou pravidla 1 2 3. Prvok 4 pomocou pravidla 1 1 4. Na výrobu prvku 5 použijeme v prvom kroku pravidlo 1 1 4 a v druhom kroku pravidlo 2 4 5. Na výrobu prvku 6 najprv použijeme pravidlá 1 2 3, 1 1 4, potom 2 4 5 a v poslednom kroku pravidlo 3 5 6.

5. Optimálne skákanie

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

Onedlho sa v ZOO konajú Kruté Skákacie Preteky. Tieto preteky sú také kruté, že sa na nich nemôže skákať hocijako. Každý súťažiaci si na začiatku pretekov tajne vyberie n rôznych dĺžok (kladné celé čísla, merané v zoometroch) d_1 až d_n a počas celých pretekov môže skákať len o tieto vzdialenosti.

Pretekárska trať je tiež dosť krutá a vyzerá nasledovne: Na jednom brehu obrovského jazera je štart. Jeden zoometer od neho je prvý kameň. Dva zoometre od prvého kameňa je druhý... Vo všeobecnosti i -ty kameň je $1 + 2 + 3 + 4 + 5 + \dots + (i - 1) + i$ zoometrov od brehu.

Všetci skáču stále jedným smerom, z i -teho kameňa na $(i + 1)$ -vý kameň. Aby boli preteky naozaj kruté, môžu sa medzi každými dvoma kameňmi najviac dvakrát odraziť od vodnej hladiny.

Vyhráva ten, kto doskáče na čo najvzdialenejší kameň.

Určite v tom skákaní máte už krutý chaos, tak uvediem príklad:

Minulý rok boli povolené dve rôzne dĺžky skokov ($n = 2$).

Opičiak Luxusko si vybral dĺžky **2 a 3**, no totálne to prehral, lebo nedokázal skočiť ani na prvý kameň.

Kačkoryba Maru sa bála, že pri dlhých skokoch zabľúdi, tak skákala len na vzdialenosti **1 a 2**. Doskákala na šiesty kameň a tam zostala.

($1 = 1$; $2 = 2$; $3 = 2 + 1$; $4 = 2 + 2$; $5 = 2 + 2 + 1$; $6 = 2 + 2 + 2$)

Andulka Anička a **havran Marek** si chceli poriadne zalietañ, zvolili si dlhé skoky **1 a 5**. Dostali sa však len na tretí kameň. ($1 = 1$; $2 = 1 + 1$; $3 = 1 + 1 + 1$)

Zajac Raf mal celkom rozumnú stratégiu, zvolil si dĺžky **1 a 4**. Skočil na rovnakom mieste ako Maru. ($1 = 1$; $2 = 1 + 1$; $3 = 1 + 1 + 1$; $4 = 4$; $5 = 4 + 1$; $6 = 4 + 1 + 1$)

Na prvom mieste však skončili **žaba Mišo** a **koza Jano**, ktorí už tento šport poznajú dlhé roky a obaja si zvolili dĺžky **1** a **3**. Dostali sa na rekordný siedmy kameň.

Úloha

Tento rok sa budú preteky konať znova a organizátori prezradili, že n bude číslo od 1 po 30. Vašou úlohou je pre každé z týchto n napísať zoznam n čísel – dĺžky skokov, ktoré by ste si zvolili, usporiadané podľa veľkosti.

Čím ďalej sa bude dať doskákať s vašou sadou, tým viac bodov za ňu dostanete.

Odvzdávanie

Namiesto zdrojového kódu posielajte na testovanie textový súbor, v ktorom bude 30 riadkov. Na i -tom riadku (číslujeme od 1 po 30 vrátane) bude i celých čísel od 1 po 10^8 – vami vybraná sada skokov pre $n = i$.

Pokiaľ pri riešení úlohy použijete nejaký program, boli by sme radi, keby ste nám ho poslali spolu s popisom.

Príklad odovzdaného súboru

```
1
1 2
1 2 3
1 2 3 4
...
Ďalších 26 riadkov
...
```

Hodnotenie

Odvzdaný súbor musí spĺňať podmienky zo zadania, inak dostane 0 bodov.

Ak je súbor správny, označme L_i číslo kameňa, kam sa dokážete dostať s i -tou sadou skokov. Za i -ty riadok odovzdaného súboru dostanete čiastkové skóre $S_i = L_i/i$. Celkové skóre bude súčtom čiastkových.

Riešiteľ s najvyšším skóre S_{\max} dostane plný počet – 10 bodov.

Každý riešiteľ, ktorý dosiahol skóre S_{moje} , dostane $10 \times S_{\text{moje}}/S_{\max}$ bodov.

Zvyšných 5 bodov môžete dostať za popis svojho riešenia. Môžete nám poslať program, ktorým ste svoje sady vypočítali a približný čas, ako dlho vám tento program bežal. Môžete pouvažovať nad tým, v čom je vaše riešenie dobré, prečo ste si ho zvolili, čo by sa ešte dalo zlepšiť.

6. Ostrieňaní zlomyseľníci

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

„Muahahahahá!“ hurónsky sa zarehotal brat Zákerný. „Ale sme ich vtedy prehrali!“

Brat Zamyslený, ktorý sa túto jeseň výnimočne nezúčastnil pravidelného druidského sympózia, sa strhol a prekvapene spýtal: „A čože ste im to, neborákom, vyviedli?“

„No, asi v polovici sympózia sme presťahovali naše táboreisko o pekný kus ďalej na sever. Uprostred noci! Mal si vidieť, ako sa tváрили po tom pochode horami, naložení batožinou...“

Aby ste boli v obraze, najstarší členovia spoločenstva druidov organizujú dvakrát ročne sympóziu pre svojich mladších kolegov. Učia ich tam rôznym figľom, delia sa s nimi o svoje skúsenosti a občas im... no... robia zle.

„Tak, tak. Na to len tak ľahko nezabudnú,“ pridal sa brat Znalecký. „Ale čo si vymyslíme na nasledujúce jarné sympóziu? Nesmieme predsa poľaviť.“

Brat Zákerný po krátkom zamyslení zvolal: „Mám to! Založíme táborisko priamo pred chatrčou niektorého z našich mladších druidov. Ten to bude mať sice hneď pod nosom, no ostatní sa poriadne nachodia.“

„Niektorí naši členovia však bývajú v odľahlých krajoch,“ ozvala sa i sestra Zblúdila, „mali by sme si výber miesta poriadne zvážiť. . . aby sa nenachodili až príliš.“

„Máš mäkké srdce; vidno, že si medzi nami ešte krátko,“ odvetil brat Znalecký. „No nech je po tvojom. Aj tak to bude najbrutálnejšie sympóziu od čias, keď nás opustil brat Zlý.“

Úloha

Chatrče druidov sú očíslované od 1 do n . V i -tej chatrči býva druid, ktorý si na sympóziu vezme batožinu s hmotnosťou w_i .

Cestná sieť je tvorená obojsmernými cestami. Pre každú dvojicu chatrčí existuje práve jedna postupnosť ciest, po ktorej sa dá medzi nimi prejsť.

Nepohodlie druida vypočítame ako súčin hmotnosti jeho batožiny a dĺžky trasy, ktorú musí prekonať, kým prejde zo svojej chatrče na miesto sympózia. *Celkové nepohodlie* vypočítame ako súčet nepohodlia všetkých druidov.

Pomôžte sestre Zblúdilej a zistíte, aké je najmenšie dosiahnuteľné celkové nepohodlie, ak za miesto konania sympózia zvolíme niektorú z chatrčí druidov.

Formát vstupu

V prvom riadku vstupu je číslo n ($1 \leq n \leq 100\,000$). Nasleduje n riadkov, i -ty z nich obsahuje číslo w_i ($0 \leq w_i \leq 1\,000$) – hmotnosť batožiny i -teho druida.

Nasledujúcich $n - 1$ riadkov popisuje cestnú sieť. Každý z nich obsahuje tri čísla a_i b_i d_i ($1 \leq a_i, b_i \leq n, 1 \leq d_i \leq 1\,000$) – cesta spája chatrče a_i a b_i a má dĺžku d_i .

Formát výstupu

Na výstup vypíšte jeden riadok obsahujúci najmenšie možné celkové nepohodlie.

Príklad

vstup

```
4
0
10
3
5
1 2 2
3 1 1
1 4 1
```

výstup

```
24
```

Najvhodnejším miestom je chatrč 2: nepohodlie druida 1 bude 0 (lebo nemá žiadnu batožinu), nepohodlie druida 3 bude $3 \cdot 3 = 9$ a nepohodlie druida 4 bude $5 \cdot 3 = 15$.

7. Obecná pošta

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

Na ďalekom východe Slovenska leží obec Klobásová s Párkami. Je to typická dedina – jedna dlhá ulica, okolo ktorej sú postavené domy a každý má svoje unikátne číslo. Vždy, keď sa v dedine postavil nový dom, dostal ľubovoľné číslo, ktoré bolo voľné. Preto sa stalo, že čísla domov nejdú v poradi, ako by ste čakali. Napríklad sa môže stať, že medzi domami č. 4 a č. 5 je dom s číslom 42.

Pred pár rokmi si tu Jano založil poštovú spoločnosť J&P.⁴ Na začiatku dediny postavil budovu pošty a odtiaľ rozvážal zásielky. Raz do týždňa nasadol do svojho auta, prešiel celú ulicu a poštu rozdal. Rýchlo však zistil, že život na dedine je drsný. Občas totiž cestu medzi niektorými domami zasypalo snehom, občas ju zablokoval dobytok... Jano potom nemohol doručiť poštu do vzdialených častí dediny, keďže sa tam nemal ako dostať.

Preto sa rozhodol požiadať o dotáciu na postavenie nových ciest s cieľom zabrániť takýmto problémom. Zistil si, medzi ktorými domami sa dajú postaviť nové cesty. Možných ciest je však veľa a peňazí málo a on by chcel vybrať čo najmenej ciest tak, aby platilo, že ak bude ľubovoľná jedna cesta zablokovaná, stále bude vedieť doručiť poštu do každého domu.

Formát vstupu

V prvom riadku vstupu je číslo n ($2 \leq n \leq 100\,000$) predstavujúce počet domov v dedine (vrátane budovy pošty, ktorá má špeciálne číslo 0). Na ďalšom riadku je n čísel x_i ($0 \leq x_i \leq 100\,000$) predstavujúce poradie domov pozdĺž ulice, čo znamená, že z domu x_i vedie priama cesta do domu x_{i+1} . Na treťom riadku je číslo m ($1 \leq m \leq 100\,000$) znamenajúce počet ciest, ktoré sa dajú postaviť. Nasledujúcich m riadkov popisuje dvojice domov a_i a b_i , medzi ktorými sa dá postaviť nová cesta. Môže sa stať, že medzi dvoma domami sa dá postaviť aj viacero ciest.

Formát výstupu

Na výstup vypíšete najmenší počet ciest, ktoré treba postaviť, aby platilo, že po odstránení ľubovoľnej jednej cesty sa stále bude dať dostať z každého domu do každého iného. Môžete predpokladať, že taká množina ciest vždy existuje.

Príklad

vstup

```
5
0 6 2 3 1
4
0 6
1 2
2 3
6 3
```

výstup

```
3
```

Jano potrebuje postaviť cesty medzi domami 0 a 6, 1 a 2, 6 a 3. Keby napríklad nepostavil poslednú cestu, tak po odstránení cesty 6 – 2 by sa z 0 nevedel dostať do domov 2, 3 a 1.

8. Obozretný tukan

kat. O; 17 b za popis, 8 b za program

Všetci podnikaví domorodci konečne odišli a pustošenie džungle sa zastavilo. Zvieratkám však naďalej zavadzali veci, ktoré domorodci za sebou nechali. Vymysleli si však dobrý systém – vždy, keď niekomu niečo zavadzalo, odhodil to o kus ďalej. Časom sa im podarilo nahromadiť všetky veci na jednu kopy uprostred veľkej lúky.

Jedného dňa si však tukan s holubom prišli zahrať na svoju obľúbenú lúku futbal a zistili, že im celkom zavadzia veľká kopa smetí. Ešteže o kúsok ďalej postávala banda kapybár, o ktorých je dobre známe, že sa radi opoja nahnitým ovocím. No a po to tukan s holubom bez problémov zaletia.

Netrvalo dlho, kým sa dohodli: Vždy, keď kapybary dostanú ovocie, odnesú niekoľko vecí (prsný počet závisí od *opojnosti* ovocia) do jazera. Nech aj ryby vedia, čo je to mať neporiadok v džungli. Tukan a holub majú v dolete niekoľko druhov ovocia, z každého druhu nekonečne

⁴Samozrejme, je to skratka pre Jano a Pošta.

veľa kusov. Ak nakoniec zostane smetí menej, než by mali kapybary podľa dohody odnieť, odnesú ich aj tak všetky.

Tukan rýchlo využil príležitosť a navrhol holubovi hru – budú sa striedať v nosení ovocia a ten, kto dosiahne ako prvý vyčistenie lúky, bude mať na začiatku futbalu loptu. Keďže tukan túto duchaplnú hru vymyslel, začína.

Holubovi ani tukanovi vôbec nevadí, že nepoznajú presný počet smetí na kope. Obaja hrajú za každých okolností optimálne (podľa svojej vtáčej intuície). To znamená, že ak existuje stratégia, ktorá hráčovi na ľahu zaručí víťazstvo (nech už bude súper ťahať ako len chce), tento hráč potiahne podľa takej stratégie.

Úloha

Tukan by rád vopred vedel, aké sú jeho šance na výhru. Správne odhadol, že smetí na kope je najviac n . Teraz potrebuje spočítať, koľko je takých čísel x z intervalu $[1, n]$ (vrátane), že pre kopy s x smetami vyhrá.

Formát vstupu

V prvom riadku je číslo n ($1 \leq n \leq 10^{18}$) – horný odhad veľkosti kopy. V druhom riadku je počet rôznych druhov ovocia v okolí k ($1 \leq k \leq 20$).

V poslednom riadku je k rôznych čísel oddelených medzerami – opojnosti jednotlivých druhov ovocia v dolete, merané v počte odnesených vecí o_i ($1 \leq o_i \leq 20$).

Formát výstupu

Vypíšte jedno číslo predstavujúce počet veľkostí kôp z intervalu od 1 po n vrátane, pre ktoré začínajúci hráč vyhrá.

Príklad

vstup

```
9
2
5 3
```

výstup

```
6
```

Tukan prehrá, len ak má kopa veľkosť 6, 7 alebo 8. Pre kopy 1, ..., 5 vyhrá prinosením ovocia 5; pre kopy 9 vyhrá s ovocím 3.

9. Trolololo

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

Známy spevák Eduard Anatoljevič Čil⁵ sa rozhodol vydať sa so svojím posledným hitom na okružnú cestu Ruskom. Začína v Moskve, pôjde na východ do Vladivostoku a potom sa bude vracat späť na západ do Moskvy. Keďže chce zarobiť čo najviac, bude sa snažiť prejsť cez čo najviac miest. Ale pretože má aj veľa nepriateľov, nechce riskovať: do žiadneho mesta (okrem Moskvy) na svojej ceste nevstúpi viac ako raz.

Úloha

Na vstupe je n miest a m obojsmerných letov medzi nimi. Nájdite najdlhšiu cestu tvaru $1, a_1, a_2, \dots, a_x, n, b_1, b_2, \dots, b_y, 1$, kde $1 < a_i < a_{i+1} < n$ a $n > b_j > b_{j+1} > 1$. Navyše každé mesto okrem 1 je v ceste maximálne raz.

Formát vstupu

⁵pozri <http://www.youtube.com/watch?v=2Z4m41njxkY>

V prvom riadku sú čísla n, m ($2 \leq n \leq 5000, 2 \leq m \leq 10000$). Potom nasleduje m riadkov popisujúcich jednotlivé lety. Každý z nich obsahuje dve čísla – mestá, medzi ktorými lieta daný let.

Formát výstupu

Vypíšte jeden riadok obsahujúci medzerou oddelené mestá na najdlhšej ceste. Ak sa cesta uskutočniť nedá, vypíšte TROLOLOLOLO. Ak je najdlhších ciest viacero, vypíšte ľubovoľnú z nich.

Príklad

vstup	výstup
5 4 1 2 2 5 3 5 1 3	1 2 5 3 1

10. Tajný kód

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

Kleofášovi sa pokazil vysávač. Tak sa vybral kúpiť si nový – ale ceny vysávačov nedávno išli hore, a keď Kleofáš videl, koľko peňazí potrebuje, skoro ho porazilo. Po dlhom zvažovaní rôznych možností, ako rýchlo prísť k veľkému množstvu peňazí, sa rozhodol, že najlepšie bude vylúpiť banku. (Je síce pravda, že lúpež banky je pomerne nebezpečná záležitosť, ale ostatnými metódami by si na vysávač nezarobil ani za celý život.)

Teraz už má Kleofáš pripravené všetko potrebné, zostáva mu len posledná vec: zistiť kód do bankového trezoru. Každý trezor má nejaké číslo, pomocou ktorého sa dá otvoriť, a to číslo je vždy v tvare p^q , kde p a q sú prvočísla. (To jest, kód od trezoru je určite nejaké prvočíсло umocnené na nejaké, možno iné, prvočíсло.) Takéto čísla sú kódy. Keď sa číslo nedá zapísať v takom tvare, nemôže to byť kód.

Kleofáš na čiernom trhu ešte k tomu zistil túto informáciu: do tej banky sa už nedávno niekto vlúpал. Kódy menšie alebo rovné ako n ani neskúšal, a aj tak ešte musel vyskúšať k kódov, kým sa dostal dnu. Kleofáš sa najprv zľakol, že bude tiež musieť skúšať k kódov, ale potom si uvedomil, že mu táto informácia už stačí na to, aby ten kód vedel presne zistiť. Len ho treba vypočítať dostatočne rýchlo, lúpež sa má čoskoro začať.

Formát vstupu

Na vstupe je jediný riadok s dvoma číslami, n ($1 \leq n \leq 10^{18}$) a k ($1 \leq k \leq 10^5$).

Formát výstupu

Na výstup vypíšte jeden riadok obsahujúci kód od trezoru tejto banky – to jest, keď zoberieme čísla, čo by mohli byť kódom a sú väčšie ako n , chceme k -te najmenšie z nich.

Príklad

vstup	výstup
25 3	49
vstup	výstup
217104242 47	229345007