



Korešpondenčný seminár z programovania

Leták zimnej časti 42. ročníka

Korešpondenčný seminár z programovania (KSP) je súťaž programátorov – stredoškolákov a mladších – pripravovaná skupinou študentov Fakulty matematiky, fyziky a informatiky Univerzity Komenského v Bratislave. Naším cieľom je zdokonaľiť žiakov v programovaní a v algoritmickom myslení.

Riešením súťažných úloh a štúdiom vzorových riešení sa zlepšíš v programovaní a naučíš sa algoritmicky rozmýšľať. Získané poznatky a skúsenosti využiješ v iných súťažiach v programovaní (napríklad pri riešení [Olympiády v informatike](#)), v bežnom živote, počas vysokoškolského štúdia, dokonca aj na prijímacích pohovoroch do zamestnania. Naši riešitelia sa každoročne zúčastňujú a úspešne umiestňujú na medzinárodných olympiádach v informatike (v Austrálii, Taliansku, Kazachstane, Taiwane, ...). Mnoho našich bývalých riešiteľov sa tiež bez ťažkostí zamestnalo v špičkových IT spoločnostiach ako Google, Facebook, ESET, ...

Ak študuješ na strednej škole a zaujíma ťa programovanie, neváhaj a zapoj sa do KSP:

Ako sa zapojiť do KSP?

- **Prečítaj** si zadania. Nájdeš ich v tomto letáku a na našej stránke <https://www.ksp.sk/ulohy>. Každý rok máme zimnú a letnú časť, obe majú dve kolá s ôsmimi úlohami.
- Teš sa, aké sú tento rok pekné úlohy.
- **Vyrieš** úlohy. Nemusíš vyriešiť všetky, nemusíš ich vyriešiť najlepšie ako sa dá. Aj za čiastočné riešenia sa dostávajú body, za každú úlohu za dá získať 0 až 20 bodov.
- Na riešenie úloh jedného kola máš približne dva mesiace a môžeš ich riešiť doma bez toho, aby si niekam cestoval. Termín odovzdania úloh je napísaný aj na našej stránke, aj v PDF zadaniach. Úlohy sa nedajú odovzdávať po termíne, takže si to, prosím, nenechaj na poslednú chvíľu.
- Úlohy rieš samostatne a neprezeráš riešenia ostatným riešiteľom. Odpisovanie riešení a prezradenie riešení pred termínom kola je porušením pravidiel KSP. Po skončení kola sa, samozrejme, o riešeníach rozprávať môžeš. :)
- **Odozdaj** riešenia úloh. Odkaz na odovzdávanie úloh nájdeš pod webovým zadaním každej úlohy alebo na stránke <https://www.ksp.sk/odovzdavanie>. Na odovzdávanie sa treba prihlásiť, aby sme vedeli, komu máme dať body.
 - Vo väčšine úloh odovzdávaš program a popis.
 - Program je hneď po odovzdaní otestovaný testovačom a hneď vidíš, koľko bodov za program máš. Program môžeš odovzdávať znova a znova, až kým nie si spokojný s výsledkom. Ak nevieš, ako majú vyzeráť odovzdané programy, pozri si <https://www.ksp.sk/odovzdavanie-programov>
 - Do popisu slovne napíšeš, ako tvoje riešenie funguje, prečo funguje a tiež odhad časovej a pamätovej zložitosti programu. Viac sa dozvieš na stránke <https://www.ksp.sk/ako-riesit>. Popis opraví a obodujú vedúci KSP po skončení kola.
- Po skončení kola si **prečítaj vzorové riešenia** úloh (veľa sa z toho naučíš), pozri svoje opravené popisy (či ti tam vedúci nenapísali nejaké poučné komentáre), pozri sa do výsledkovky a **teš sa**, koľko máš bodov. Vo výsledkoch sa hodnotí samostatne letná a zimná časť. V každej časti je dôležitý celkový súčet bodov.
- Prečo sa máš tešiť z bodov? Čítaj ďalej.

Čo môžem vyhrať?

- Okrem neoceniteľných vedomostí, skúseností a zručností, ktoré získaš pri riešení semináru, môžeš vyhrať množstvo skvelých vecí.
- Všetci víťazi od nás dostanú **vecné ceny**.
- Pre 36 najlepších riešiteľov organizujeme každoročne dve týždenné **sústredenia**. Sústredenie je niečo ako tábor, na ktorom spoznáš nových priateľov s podobnými záujmami, naučíš sa čosi viac nielen o programovaní a zažiješ kopec zábavy. Sústredenia sú fakt skvelé akcie, najmä, keď ich organizuje Trojsten.

- Aby ste sa mohli pochváliť ostatným, akí ste šikovní, víťazom všetkých levelov udelíme a pošleme **diplomy**.
- Aj keď sa nedostaneš medzi víťazov, stále môžeš byť úspešným riešiteľom. Úspešný riešiteľ je ten, kto získal aspoň polovicu bodov počas celej časti (letnej, či zimnej).

Pravidlá a levely

Počnúc tridsiatym piatym ročníkom rušíme staré kategórie a prechádzame na nový systém *levelov*.

Každý riešiteľ má level, číslo od 1 po 4. Noví riešitelia začínajú na leveli 1 a pokiaľ sa im v riešení darí, level im postupne rastie. Svoj level si môže každý riešiteľ pozrieť na našej stránke. Riešiteľom s levelom L sa započítavajú body len za úlohy s číslami L až 8.

Vo výsledkových listinách (<https://www.ksp.sk/vysledky>) sa každému riešiteľovi počíta **5 najlepšie vyriešených úloh**. Celkovo sa dá za časť (dve kolá) získať 200 bodov. Riešitelia, ktorí sa v nejakej výsledkovke umiestnili na jednom z prvých dvoch miest a majú aspoň 150 bodov sú **víťazi**. Najlepších 36 riešiteľov pozývame na sústreďenie.

Podrobnejšie pravidlá si môžete prečítať na <https://www.ksp.sk/pravidla>.

Registrácia

Pred odovzdaním riešenia je potrebné sa zaregistrovať na našej webstránke a vyplniť požadované kontaktné údaje. Odporúčame sa zaregistrovať aspoň pár dní pred odovzdávaním riešenia (pre prípad, že by ste mali počas registrácie nejaké problémy).

Účastou 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.



Úlohy 1. kola zimnej časti

Termín odoslania riešení tohto kola je pondelok **21. októbra 2024**. Doprogramovávanie končí v pondelok 4. novembra 2024.

1. Hlúpa intergalaktická pošta!

12 b za popis, 8 b za program

Náš kamarát Miško z planéty “Flexi 128” chcel podporiť vymýšľanie tohto kola, a preto s nami chcel zdieľať jeho knižnicu hier na intergalaktickom obchode Para™. Nevedel nám však poslať jeho heslo, kvôli tej hlúpej intergalactickej pošte! Našťastie, jeho heslo má iba jediný znak v kódovacej sústave UTF-4096, čo je štandardizované kódovanie v širokej galaxii. A máme iného kamaráta, ktorý si pamätá ako vzniklo - je to predsa iba počet krokov, ktoré potrebuje na zadanie pin kódu na svojich smart StarFruit™ hodinkách. Celkom ľahko sa používajú - stačí iba otáčať vonkajším okruhom, ako trezorom a vždy, keď je na čísle, ktoré by chcel zadať, stlačí displej. Jediný problém je, že jeho pin kód vie byť veľmi dlhý. Jazyk C** alebo Cobra sa už takýmito ľahkými problémami nezaoberajú. Pomôžte nám!

Úloha

Naprogramujte program, ktorý zistí minimálny počet krokov, ktoré potrebujeme na zadanie Miškovo pin kódu na jeho hodinky.

1	2	3	4	5	6	7	8	9	0
---	---	---	---	---	---	---	---	---	---

Takto vyzerá pásik na hodinkách. Treba si uvedomiť, že sa vieme pohnúť z 0 na 1 a naopak. Každý pohyb doľava či doprava sa ráta ako jeden krok, a taktiež musíme zaradiť stlačenie displeja ako jeden krok. Treba ešte spomenúť, že začíname “ukazovátkom” na 1, a nemusíme sa vrátiť na pôvodnú pozíciu po stlačení.

Formát vstupu

V prvom riadku vstupu je reťazec dĺžky n ($1 \leq n \leq 10^6$) - Miškovo pin kód. Jeho pin kód bude obsahovať iba znaky od 0 po 9.

Pre jednotlivé sady platia takéto obmedzenia:

Sada	1	2	3	4
$1 \leq n \leq$	20	1000	10^4	10^6

Formát výstupu

Vypíšte na jeden riadok minimálny počet krokov, ktoré Miško potrebuje na zadanie pin kódu - teda heslo od jeho Para™ účtu.

Príklad

vstup	výstup
<input type="text" value="820"/>	<input type="text" value="12"/>
<i>Naša cesta je nasledovná 1-8-klik-2-klik-0-klik. Treba si uvedomiť že cesta z 1-8 ide $1 \Rightarrow 0 \Rightarrow 9 \Rightarrow 8$, takže 3 kroky</i>	
vstup	výstup
<input type="text" value="8573"/>	<input type="text" value="16"/>

2. Vladkova hra

12 b za popis, 8 b za program

Vladko sa veľmi rád hrá počítačové hry. Naposledy si cez letné zľavy v intergalaktickom obchode Para™ kúpil vesmírnu ploštinovku. Narazil však na náročný level, v ktorom sa nachádza niekoľko políčok vedľa seba, na

niektorých sa nachádzajú plošinky. Vladko začína úplne vľavo, cieľ levelu je úplne vpravo. Hýbať sa môže iba doprava. Vždy, keď sa postaví na políčko s plošinkou, môže sa posunúť doprava a plošinka sa za ním rozpadne. Ak sa postaví na políčko bez plošinky, spadne do lávy (musí začať znovu) a na políčku sa objaví nová plošinka.

Úloha

Vladko chce zistiť, na koľko najmenej pokusov vie tento level dokončiť.

Formát vstupu

V prvom riadku vstupu je číslo n ($1 \leq n \leq 60$), ktoré určuje počet políčok vedľa seba.

Nasleduje jeden riadok, na ktorom je medzerou oddelených n čísiel 0 alebo 1, ktoré určujú, či sa na danom políčku nachádza plošinka, alebo nie.

Sada	1, 2	3, 4	5, 6	7, 8
$1 \leq n \leq$	10	30	60	60

Formát výstupu

Vypíšte jeden riadok s najmenším počtom pokusov, s ktorým sa dá tento level vyhrať. Pozor, výstupné číslo sa ti nemusí zmesiť do premennej typu `int` v C++.

Príklad

vstup	výstup
<pre>3 0 0 1</pre>	<pre>3</pre>

Potrebuje 3 pokusy: $0\ 0\ 1 \rightarrow 1\ 0\ 1 \rightarrow 0\ 1\ 1 \rightarrow 1\ 1\ 1$

3. Intraplanetárne čížmy

12 b za popis, 8 b za program

Jožko Číž už dlhé roky prebýva na istej planétke. Zo začiatku sa mu všetko zdalo zaujímavé a bol celý nadšený. Časom však nadšenie opadlo a začínal sa trochu nudiť. Planétka má slabú gravitáciu a dobre sa tu skáče. To však Číž už dobre pozná. Tentoraz však uvidel čížmy. Nie obyčajné čížmy. Vesmírne skákacie čížmy! Pozrel sa ďalej a uvídal ďalšie. A ďalšie. A ďalšie! Videl celú cestu, na ktorej bolo veľa rôznych skákacích čížiem.

Rozhodol sa, že sa vydá na túru na koniec tejto cesty. Samozrejme sa mu nechce chodiť a prezúvať čížmy po tom, čo sa vybijú, je pre energeticky úsporného Číža neprípustné. S každými čížmami dokáže preskákať inú vzdialenosť dokedy sa vybijú. Zaujímalo by ho, koľko najmenejkrát sa musí prezúť na to, aby sa dostal na koniec cesty.

Úloha

Máme cestu danej dĺžky na ktorej začiatku sa nachádza Číž. Na niektorých pozíciách cesty sú čížmy, do ktorých sa môže Číž prezúť, ak je na ich pozícií. Každé čížmy majú inú vzdialenosť, ktorú je nimi možné preskákať. Vašou úlohou je zistiť, koľko najmenej krát musí čížmy prezúť aby sa dostal na koniec cesty, bez toho aby chodil, alebo zistiť, že to možné nie je.

Formát vstupu

V prvom riadku sú tri celé čísla n, d, b . Číslo n ($1 \leq n \leq 500\,000$) udáva počet čížiem na ceste. Číslo d ($2 \leq d \leq 10^9$) udáva dĺžku cesty a b ($1 \leq b \leq 10^9$) udáva vzdialenosť ktorú je možné prejsť s čížmami, ktoré má Číž už obuté.

Následuje n riadkov popisujúcich čížmy na ceste. Na i -tom riadku, sú celé čísla A_i ($1 \leq A_i < d$) a B_i ($1 \leq B_i \leq 10^9$) popisujúce pozíciu čížiem na ceste a vzdialenosť ktorú je s nimi možné prejsť.

Je zaručené, že hodnoty A_i sú zoradené vzostupne. Teda čížmy sú vypísané postupne v poradí, ako sa nachádzajú na ceste. Na jednej pozícií sa tiež nachádzajú vždy najviac jedny čížmy.

V jednotlivých sadách platia aj nasledujúce obmedzenia pre n :

Sada	1	2	3	4
$1 \leq n \leq$	15	2 000	100 000	500 000

Formát výstupu

Vypíš jeden riadok a v ňom jedno celé číslo. Najmenší možný počet prezutí číziem potrebný na prejde nie na koniec cesty. Ak cestu nie je možné prejsť pomocou číziem, vypíšete -1 .

Príklady

vstup	výstup
<pre>5 20 1 1 2 2 4 3 1 5 16 7 20</pre>	<pre>3</pre>

Musíme zobrať prvé číziem a následne sa najviac oplatia druhé. Potom vieme zobrať štvrté (na pozícii 5), s ktorými sa vieme dostať až na koniec cesty.

vstup	výstup
<pre>4 10 2 1 2 2 2 6 4 8 1</pre>	<pre>-1</pre>

Síce vieme zobrať prvé, alebo druhé číziem, ale potom sa už k ďalším nevieme dostať a ani dôjsť na koniec cesty. Odpoveď je teda -1 .

4. Existenčná Kríza

10 b za popis, 10 b za program

MisQo nenosil respirátor a dostal existenčnú krízu. Teraz mu nedajú spať otázky ako napríklad:

- “čo je to TLB a aké má využitie?”,
- “má jazyk generovaný turingovym strojom danú netriviálnu vlasnosť?”,
- “aký bol tvoj názor na voľný program? bolo ho málo alebo veľa?”
- “kedy vyjde nový Taylor Swift album?”
- a, samozrejme, “aká je odpoveď na otázku Života, Vesmíru a Všetkého?”.

Tak sa raz MisQo jedného dňa prechádzal po Ikei, dumajúc nad týmito existenčnými otázkami, keď tu zrazu začul tiché “Pšt!”¹. Obzrel sa, a uvidel záhadnú postavu v kabáte, ktorej nebolo vidno do tváre. Teda možno to bolo tým, že mala len pár centimetrov. Pisklavý hlas oznámil “tento počítač má všetky odpovede, ktoré hľadáš...”. MisQo s nadšením pochybný obchod prijal a domov priniesol masívnu krabicu. Keď ju však doma otvoril, čakalo ho prekvapenie. V krabici bolo iba zrkadlo a kryptický nápis “Kľúčom si ty sám...”. MisQo si to, samozrejme, vyložil tak, že všetky súčiastky budú nechcené veci, ktoré nájde vo svojom byte.

Po prehrabaní zopár kútov našiel nasledovné artefakty, z ktorých každý má nejakú ikonickú funkciu:

1. starožitný železák dávkoč (z), ktorý už dávno nemá železo. Jeho funkcia je taká, že vždy vracia 0.
2. inhalátor (i) plný veľmi prílnavých trblietok, ktoré skrásia čokoľvek, čoho sa dotknú. Jeho funkcia je taká, že čokoľvek je na vstupe, vráti o 1 zvýšené.
3. nekonečná sada (s) zberačov odpadkov všetkých možných dĺžok. Pre každé prirodzené číslo n sa nájde nejaký zberač s_n , ktorého funkcia je taká, že vráti n -tú hodnotu zo svojho vstupe.
4. odniekiaľ zo steny vytiahnutú kovovú trubku (K), do ktorej sa dá nasypať niekoľko iných artefaktov. Jej funkcia je taká, že na svojich vstupoch spustí všetky tieto artefakty, až na prvý, ktorý spustí až na konci, na výsledkoch všetkých predošlých.
5. prastarý rotor (R), na ktorý sa dá pripievať nejaký iný artefakt, aby svoju funkciu vykonal niekoľkokrát za sebou. Jeho funkcia je taká, že zo vstupe prečíta číslo, a toľkokrát spustí na zvyšku vstupe artefakt, ktorý sa na ňom točí.

Teraz by MisQo rád zostrojil počítače, ktoré mu dajú odpovede na jeho kritické otázky. Samozrejme, s takto skromnými materiálmi to nie je vôbec triviálne. MisQovi sa veľmi nechce, a tak je na vás, aby ste pre neho tieto počítače zostrojili...

¹<https://susi.trojsten.sk/pst/>

Úloha

Táto úloha sa rieši interaktívne a môžete ju riešiť [tu](#)². Tam aj nájdete podrobnejšie pokyny k tomu, ako fungujú jednotlivé artefakty a ako sa z nich dajú skladať počítače.

K tejto úlohe píšete aj popis. Pre každú podúlohu by mal popis obsahovať maximálne 5 viet, kde opíšete, čo ste vlastne urobili a prečo to funguje a screenshot programu. Nemusí obsahovať žiadnu analýzu zložitosti ani nič podobné.

Každá podúloha má 1 bod za program a 1 bod za popis. Body za program sa ti pripočítajú automaticky, keď odovzdáš správny program, body za popis dostaneš niekedy po konci kola.

V prípade akýchkoľvek otázok ohľadom tejto úlohy [napíšte Andrejovi](#)³.

5. Zašifrovať budeme

12 b za popis, 8 b za program

Ahojte moji bratia ktorým sa otvorili oči, ako už všetci isto vieme, mimozemšťania sa snažia získať našu planétu. Našťastie, naša verná kamarátka M.i.š.k.a - Mimoriadne Iniciatívna Šifrovateľská Kamarátka (Asi) - našla spôsob, ako sa brániť. Vymyslela, ako zašifrovať všetky naše dôležité informácie, veľmi zložitým a komplexným spôsobom, že ich mimozemšťania nikdy nerozlúšia. Ako to spravila? Funguje to tak, že ku každému písmenku vymyslela náhodné slovo a následne každé písmenko z našej správy zamenila za toto slovo. Ale potom si povedala, že toto by tí ufóni mohli veľmi ľahko dekodovať. Preto to spravila znovu. Každé písmenko z novej správy vymenila za slovo jemu prislúchajúce, rovnako ako prvý krát. Stále sa jej ale správa nezdała dosť zakódovaná, tak pre istotu tento proces zopakovala ešte N -krát. Teraz je už správa veľmi dobre zakódovaná, ale keď ju posielala našim spojencom, mimozemšťania nám ukradli jedno písmenko z tejto zakódovanej správy. Pomôžte nám zistiť, ktoré písmenko nám ukradli.

Úloha

Na vstupe dostanete ku každému písmenku z abecedy priradený string nejakej náhodnej dĺžky. Taktiež tam dostanete aj pôvodnú správu a počet N (koľko krát naši vedci vymenili každé písmenko zo správy za jemu prislúchajúci string). Na záver ešte dostanete aj číslo K . Vašou úlohou je vypísať K -te písmenko z výslednej správy. Ak výsledná správa je kratšia ako K tak vypíšte "Neexistuje". Zároveň existuje aj číslo D , ktoré udáva maximálnu dĺžku stringu priradeného písmenku z abecedy.

Formát vstupu

V prvom riadku vstupu je string, ktorý vedci chcú zakódovať. V tejto úlohe používame iba malé písmenká anglickej abecedy. String bude mať dĺžku aspoň 1 a najviac 10^6 . V druhom riadku sú čísla N udávajúce koľko krát sa má každé písmenko zmeniť za jemu prislúchajúci string a K udávajúce koľkaté písmenko z výsledného reťazca chceme vedieť. Následuje 26 riadkov kde na i -tom riadku je string priradený písmenku i -temu z abecedy (na prvom riadku ku a -čku na druhom b -čku atď). V každom riadku sa nachádza alespoň 1 znak. Jediný prípad, kedy v riadku nedostanete písmenko malej abecedy je, keď sa tam bude nachádzať $\#$. Tento znak znamená, že za písmenko vymieňame vždy prázdny string.

V jednotlivých sádach platia nasledujúce obmedzenia (kde D reprezentuje aký najdlhší môže byť string ktorým nahrádzame jednotlivé písmenká):

Sada	1	2	3	4
$1 \leq N \leq$	10	25	10^4	10^5
$1 \leq D \leq$	1 000	10^4	1 000	1 000
$1 \leq K \leq$	1 000	10^6	10^6	10^8

Zároveň ešte pre jednotlivé sady platí že:

Sada	Obmedzenie
1	$D^N \leq 10^6$
2	$D^*N \leq 1\,000$
3	$D^*N \leq 10^5$
4	$D^*N \leq 10^5$

²<https://ksp.sk/specialne/ksp/41/1/>

³<mailto:andrej.lackovic+existencna.kriza@trojsten.sk>

Formát výstupu

Vypíšte buď K -te písmeno z výslednej správy (čísľujúc od 1) alebo “Neexistuje” ak je výsledná správa kratšia ako K .

Príklady

vstup

```
ab
1 4
abb
cde
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
```

výstup

```
c
```

z a-čka sme spravili “abb” a z b-čka “cde”, čo znamená, že z výsledného stringu “abcde” chceme 4. písmenko, čo je c

vstup

```
abc
3 14
a
cdc
dd
eee
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
```

výstup

```
Neexistuje
```

z nášeho slova abc sa po prvej zámene stalo "acdcdd", po druhej sa z neho stalo "addeeeddeeeeee" a po tretej "aeeeeeeeeeee", keďže e-čka meníme na prázdny string. A keďže sa pýtame na pozíciu, ktorá vo výslednom stringu neexistuje, tak vypíšeme "Neexistuje"

vstup

```
zwx
4 66
qoyh
bfd
m
dsha
gnif
fyz
#
t
oeea
#
y
f
j
gr
ugf
jal
fei
#
jt
atvw
cj
jp
kpw
b
ax
ma
```

výstup

```
a
```

vývoj stringu: zwx -> makpwb -> jqoyhyjalkpwbfd -> feiugfartaxqoyhfyzjalkpwbfdfyzdsha -> fyzgnifoeecjfy-zqoyhbatvwqoyhbfeiuqfartfyzaqoyhfyzjalkpwbfdfyzdshafyzaxmadshajttqoyh

6. Diskrétny Banket

12 b za popis, 8 b za program

Okäň “Bzučiak” Hruškový, zákerný zločinec známy celou galaxiou, už má vyhladený blištivý cieľ svojej najnovšej lúpeže. Kdesi v bode v nekonečne sa totiž nachádza legendárna planéta menom Al-Jabr, na ktorej sídli mocný panteón známy ako Okruh Celých Čísel. Bzučiak plánuje stať sa lupičom lupičov, spáchať krádež krádeží - ukradne identitu identity. Teda, povedané ako človek, ukradne identitu čísla Jeden. Na to sa vláme na prestížny diskretný banket, ktorý Jednotka pravidelne organizuje...

Na diskretný banket Jednotka okrem seba pozve ešte aj nejakú konečnú množinu kladných ⁴ celých čísel. Vrátnik dostane krásne očíslovaný zoznam, na ktorom sú všetky tieto čísla vypísané v rastúcom poradí ⁵. Vyzerá to napríklad takto: 1. 2 2. 3 3. 7 4. 9 5. 10 6. 12 7. 13

Môžete si všimnúť, že to je mierne mätúci formát. A vrátnik s ním má problém tiež! Podľa protokolu by sa malo prichádzajúce číslo preukázať dokladom totožnosti, na čo vrátnik skontroluje, či je na zozname, a ak áno, dá mu menovku s jeho menom a vpustí ho na banket. Avšak vrátnik tomuto protokolu zle porozumel, a na menovku dá namiesto jeho názvu jeho poradové číslo na zozname. Teda ak by za vrátnikom so zoznamom vyššie prišlo číslo 10, dostane menovku s číslom 5, pretože je v zozname piate.

Bzučiakov plán je veľmi jednoduchý, podľa nasledovného krásne očíslovaného zoznamu: 1. Vyhladiť si vhodné celé ciele 1. Ukradnúť identitu niektorého z nich 2. Podstrčiť vrátnikovi vhodný falošný zoznam hostí 3. Vstúpiť na banket so svojou novoukradnutou identitou 4. Dostať menovku (vďaka chybe v systéme s nižším číslom) 5. Vstúpiť znovu na banket, preukázajúc sa novozískanou menovkou 6. Opakovať kroky 4 – 5 kým sa bude dať 7. Získať menovku s číslom 1 8. Ujsť na Vesmírne Belize 9. Zo všetkých zločincov, ktorí sa kedy v galaxii činili, byť 1 10. Kúpiť si zvlhčovač vzduchu

Prvý krok už splnil - má zoznam čísel, ktorých identitu je schopný odcudziť. Teraz potrebuje vašu pomoc

⁴Nula je všadeprítomná a netreba ju nikam pozývať, no a záporné čísla... Povedzme, že ich prítomnosť nie je veľmi pozitívna...

⁵Samozrejme, čísla rastú na sile, čím bližšie k Nule sú, preto je zoznam vlastne zoradený podľa klesajúcej dôležitosti hostí.

⁶, aby sa rozhodol, ktorú z počiatočných identít sa mu oplatí ukradnúť najviac. Musí totiž myslieť aj na nasledujúci krok, a podstrčiť vyhovujúci zoznam hostí. Navyiac je tu ešte problém s tým, že niektoré čísla vyzerajú veľmi ikonicky a ani Bzučiak, majster prestrojenia, nemusí byť schopný ich výzor napodobniť... Pre každú možnosť počiatočnej identity Bzučiakovi povedzte, koľko existuje zoznamov spĺňajúcich tieto (krásne očíslované) podmienky: 1. Aby to nebolo podozrivé, Bzučiakova pôvodná identita musí byť najvyšším (posledným) číslom v zozname. 2. Opakovaným nahrádzaním čísla jeho poradím v zozname sa po konečnom počte krokov dostaneme na číslo 1. 3. Ani jedno z čísel navštívených takýmto skákaním nepatrí medzi tzv. bezpečné čísla, ktoré sú zadané na vstupe.

Úloha

Na vstupe je zadaných niekoľko bezpečných čísel a niekoľko počiatočných čísel. Pre každé počiatočné číslo a vypíšte počet rôznych podmnožín Z množiny $2, 3, \dots, a$ takých, že vzostupným zoradením a očíslovaním (od 1) prvkov Z sa opakovaným nahrádzaním čísla (počnúc od a) jeho poradovým číslom dostaneme k číslu 1 bez toho, aby sme prešli bezpečným číslom, alebo číslom, ktoré nie je v Z .

Keďže počty možností môžu byť fakt veľké, vypíšte iba ich zvyšok po delení číslom $10^9 + 7$

Formát vstupu

V prvom riadku vstupu sú medzerou oddelené dve čísla r a k ($0 \leq r \leq 10^5$, $1 \leq k \leq 10^5$) - počet bezpečných čísel a počet počiatočných čísel.

Na nasledujúcich r riadkoch je na každom jedno celé číslo x_i - toto sú bezpečné čísla. Môžete predpokladať, že sú zoradené od najmenšieho, a žiadne dve sa nerovnajú.

Na k zvyšných riadkoch je po jednom celom čísle a_i - všetky počiatočné čísla. To isté počiatočné číslo sa môže vyskytnúť aj viackrát.

V jednotlivých sadách platia nasledujúce obmedzenia:

Sada	1	2	3	4
$1 \leq r, k \leq$	10^5	10^5	10^5	10^5
$2 \leq x_i, a_i \leq$	42	10	50	200

V sade 1 dokonca platí, že bezpečné nie sú žiadne čísla, teda $r = 0$.

Formát výstupu

Pre každé z k počiatočných čísel vypíšte jeden riadok s jediným číslom - počtom vyhovujúcich zoznamov pre dané počiatočné číslo.

Príklady

vstup

```
0 3
2
5
10
```

výstup

```
1
5
77
```

Pre 2 je odpoveď zrejmalá - existuje len jedna prípustná množina, a 2 je v nej najnižším číslom. 5 vyhovujú nasledovné množiny: $\{2, 3, 4, 5\}$, $\{2, 3, 5\}$, $\{2, 5\}$, $\{3, 4, 5\}$ a $\{5\}$.

vstup

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

výstup

```
0
2
23
```

Keďže 2 je bezpečným číslom, začať plán ním by okamžite zlyhalo. 5 vyhovujú tie množiny ako v minulom prípade, ale nesmie v nich byť navštívené 2 ani 4 (môžu byť v zozname, len im nesmieme vziať identitu): $\{3, 4, 5\}$ a $\{5\}$.

⁶Áno, budete akomplicom. Nemáte však na výber, pokiaľ nebudete súhlasiť, Bzučiak na galaktický internet vypustí fotky vašeho prvého programu a vaša reputácia bude zdevastovaná.

7. Nevítaný hosť s chlpatým kožucho

15 b za popis, 10 b za program

Bzučiakov zdanlivo jednoduchý plán sa nakoniec nečakane skomplikoval. Ó nie nie, celú krádež a únik zvládol triviálne! Hoci sa mu podarilo infiltrovať diskrétny banket, ukradnúť identitu a dokonca prejsť cez pasovú kontrolu na Vesmírnom Belize so zabudnutým pasom, jeho fatálnou chybou bolo nechať si kúpu zvlhčovača vzduchu na poslednú chvíľu.

Na Vesmírnom Belize sa zvlhčovače vzduchu zháňajú ťažšie než identita samotnej Jednotky. A o jeho preferovanej luxusnej značke môže Bzučiak len snívať. Jediné, čo sa mu podarilo získať, bol zjavne použitý zvlhčovač od pofidérneho predajcu vedľa smetiakov za benzínkou – a to ešte za premrštenú cenu!

S ťažkým srdcom sa Bzučiak vrátil do hotelovej izby, odhodlaný tešiť sa aspoň z toho mála, čo má. Jeho tešba však trvala len krátko. Ešte predtým, ako stihol zvlhčovač zapnúť, si všimol, že s ním niečo nie je v poriadku. Po otvorení zistil, že vnútro prístroja je celé zelené a ochlpatené – vesmírna pleseň!

Bzučiak stál pred dilemou. Zohnať náhradný zvlhčovač by bolo takmer také náročné ako vykonať celú predchádzajúcu lúpež. Na druhej strane, používať plesnivý zvlhčovač by bolo mimoriadne zdraviu škodlivé. Po chvíli váhania sa rozhodol pre kompromis – postaví zvlhčovač na balkón, nechá naň svietiť žeravé slnká trojhviezdneho systému Belize a fúkať slaný morský vzduch a hádam tá pleseň skape. V takýchto nepriaznivých podmienkach predsa nemôže dlho prežiť...

Úloha

Povrch zvlhčovača vzduchu predstavuje (z pohľadu plesne nekonečnú) mriežku. Na niektorých políčkach mriežky sa na začiatku nachádza pleseň. Zhora na mriežku praží horúce slnko, zľava fúka vysušujúci vietor. Kvôli týmto nepriaznivým podmienkam pleseň postupne vymiera, no predtým, než úplne zanikne, sa pokúsi o posledný akt sebazáchovy.

Kolónia plesne sa správa podľa nasledujúcich pravidiel:

1. Ak na plesnivé políčko priamo pôsobí aj slnko aj vietor (t.j. nemá plesnivých **susedov** ani zhora, ani zľava), život je pre ňu príliš ťažký a pleseň na tomto políčku v nasledujúcej sekunde zahynie.
2. Ak na prázdne políčko priamo nepôsobí ani slnko, ani vietor (t.j. má plesnivých **susedov** zhora aj zľava), vznikajú na ňom ideálne podmienky a v nasledujúcej sekunde sa tam pleseň rozšíri.
3. Všetky ostatné políčka zostávajú v nasledujúcej sekunde nezmenené.
4. Všetky zmeny v rámci jednej sekundy sa dejú súčasne.

Kolóniu plesne vieme reprezentovať ako množinu obdĺžnikových plôch. Plesnivá oblasť so súradnicami (x_1, y_1, x_2, y_2) obsahuje všetky políčka súradníc (x, y) takých, že $x_1 \leq x \leq x_2$ a $y_1 \leq y \leq y_2$. Plesnivé plochy sa môžu prekrývať, avšak v tom prípade je na danom políčku pleseň len raz.

Otázkou teda zostáva, pri počiatkovej konfigurácii plesne, koľko sekúnd potrvá kým všetka pleseň zahynie a Bzučiak bude môcť konečne začať používať svoj zvlhčovač?

Formát vstupu

Na prvom riadku vstupe je číslo T ($1 \leq T \leq 100$) udávajúce počet testovacích prípadov. Nasleduje T testovacích prípadov, pričom každý z nich je zadaný nasledovne:

Na prvom riadku je číslo N ($1 \leq N \leq 50\,000$) udávajúce počet plesnivých plôch. Nasleduje N riadkov, v každom z nich sú štyri celé čísla x_1, y_1, x_2 a y_2 ($1 \leq x_1 \leq x_2 \leq 10^9, 1 \leq y_1 \leq y_2 \leq 10^9$) udávajúce súradnice plesnivej obdĺžnikovej plochy.

Súradnice rastú smerom dole a doprava.

Nech $\sum N$ je súčet N pre všetky testovacie prípady v jednom vstupe. Vstupy úlohy budú rozdelené do 5 sád, v ktorých platia nasledujúce obmedzenia:

1. $T \leq 20, N \leq 10, x_2, y_2 \leq 100$
2. $\sum N \leq 1\,000, x_2, y_2 \leq 10^6$, súčet plôch všetkých obdĺžnikov $\leq 4 * 10^6$
3. $N \leq 2\,500, \sum N \leq 5\,000$
4. $\sum N \leq 50\,000$, plochy sa neprekrývajú
5. $\sum N \leq 50\,000$

Formát výstupu

Pre každý testovací prípad vypíšte jedno číslo – počet sekúnd, ktoré potrvá, kým všetka pleseň zahynie.

Príklady

vstup

```
3
1
1 1 5 1
1
1 1 1 5
1
1 1 3 3
```

výstup

```
5
5
5
```

vstup

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

výstup

```
6
```

Nech A, B, C sú plesnivé oblasti súradníc $(4, 1, 4, 1)$, $(2, 2, 3, 2)$ a $(1, 2, 1, 4)$. Nech P je pleseň ktorá voči minulej sekunde prežila a n je pleseň ktorá sa narodila. V nasledujúcich sekundách bude pleseň vyzerat takto:

...	A	1s	2s	3s	4s	5s	6s
CBB.	- \	.PPn	- \	..PP	- \	...P	- \	- \	- \	
C...	- /	Pn..	- /	.Pn.	- /	..Pn	- /	...P	- /	- /	
C...		P...		Pn..		.Pn.		..Pn		...P		

vstup

```
1
4
49 8 66 93
62 4 67 99
38 61 58 81
58 2 60 66
```

výstup

```
110
```

Keď sa pekne spýtate ChatGPT-4o, tak vám tento vstup nakreslí. Na takéto veci je AI veľmi fajn, avšak nepoužívajte ju priamo na riešenie a programovanie úloh.

Poznámka

Bol by som rád keby ste túto úlohu skúsili vyriešiť a naprogramovať (na čo najviac bodov). Aj keď je to sedmička, nie je mimoriadne ťažké získať viac ako polovicu bodov. Ak sa úlohu pokúšate riešiť na plný počet bodov, tak odporúčame programovať v $C++$ a popis napísať poriadne a zrozumiteľne.

8. Aquamarínové guličky

12 b za popis, 8 b za program

Malá T našla niekoľko modrých guličiek a nanukových paličiek, tak ich začala lepíť dokopy a staviať z nich. Začala tým, že ich pospájala tak, aby každá nanuková palička spája dve guličky, avšak jedna gulička môže byť pripojená na ľubovoľne veľa paličiek. Tiež platí, že celé dielo je súvislé, ale ak by sme ľubovoľnú paličku zobrali, rozpadlo by sa na dva kusy.

Po chvíli si však malá T uvedomila, že jej výtvor je príliš veľký a ťažko sa jej prenáša, takže chcela by ho zmenšiť. Existuje však len jeden spôsob, ako je ochotná zmenšiť svoj výtvor – zoberie jednu guličku preč a spolu s ňou aj všetky paličky, ktoré boli k nej prilepené. Je možné, že dielo sa následne rozpadne na niekoľko kusov, čo nie je dobré, preto by T chcela pridať niekoľko nových paličiek spajajúcich dve guličky tak, aby výtvor bol znovu súvislý. Dostane tak útvar, ktorý má o jednu guličku menej. Nie je to však také jednoduché. Keďže narábanie s lepiacou pištoľou je nebezpečné, bolo by dobré tých nových paličiek pridať čo najmenej. Navyše guličky sú rôznych odtieňov modrej a aby bol výsledný útvar pekný, bolo by vhodné, keby nové paličky spájali guličky s podobnými farbami.

Úloha

Malá T má n guličiek rôznych odtieňov modrej. T očíslovala guličky od najsvetlejšej po najtmavšiu od 1 po n . Následne ich pospájala paličkami tak, že ak si predstavíme guličky ako vrcholy a paličky ako hrany medzi nimi,

dostaneme strom. Teraz uvažuje nad tým, že jednu z guľičiek spolu s paličkami, na ktoré je prilepená, odstráni. Následne chce znovu pozliepať dokopy všetky časti diela (až na odstránenú guľičku) čo najmenej paličkami tak, aby bolo znovu súvislé. Kontrast novovytvoreného guľičkového diela je súčet rozdielov (v absolútnej hodnote) medzi číslami novospájaných dvojíc guľičiek. T by chcela dosiahnuť čo najmenší kontrast. Hľadá teda spôsob, ako znovu spájať guľičky tak, aby v prvom rade urobila čo najmenej spojov a spomedzi všetkých takýchto spôsobov hľadá ten, kde nové paličky vytvárajú dokopy čo najmenší kontrast.

Nie je si však istá, ako toto urobiť, a preto necháva túto úlohu na vás, šikovných programátorov. A keďže nevie ešte ktorú guľičku odstráni, chce, aby ste vypočítali minimálny počet potrebných paličiek a minimálny možný kontrast pre takéto spájanie pre každú možnú odstránenú guľičku. A ideálne by bolo, keby ste jej aj našli návod, ako pozliepať útvar, aby tieto hodnoty dosiahla. Ak existuje viacero riešení s rovnakým kontrastom a počtom pridaných paličiek, vypíšete akékoľvek.

Formát vstupu

V prvom riadku vstupu je číslo t ($1 \leq t \leq 10^5$) udávajúce počet útvarov, pre ktoré máte úlohu riešiť.

Potom nasleduje t popisov stromov. V prvom riadku každého popisu je n ($1 \leq n \leq 10^5$), počet guľičiek a nasleduje $n - 1$ riadkov popisujúcich dvojice guľičiek, ktoré sú spojené paličkou. Guľičky sú očíslované od 1 po n podľa ich farby, takže kontrast guľičiek s číslami a a b je $|a - b|$.

V jednotlivých sadách platia nasledujúce obmedzenia:

Sada	1, 2	3, 4
$1 \leq n \leq$	1000	10^5

Za každú sadu sa dajú získať 2 body. Sady 1 a 3 sú špeciálne – na ich vyriešenie stačí len nájsť minimálny počet paličiek a minimálny dosiahnuteľný kontrast po odstránení každého vrcholu, ale nemusíte nájsť správny návod na dosiahnutie tohto minima.

Formát výstupu

Pre každý z t problémov treba vypísať n popisov riešení, jeden pre odstránenie každého z vrcholov. Keďže T ešte nevie ktorú guľičku odstráni, chcela by vedieť ako postupovať, bez ohľadu na to ktorý si vyberie. Vypíšete preto postupne riešenia pre prípad že odstráni prvú, druhú, atď. až po n -tú guľičku.

V prvom riadku popisu riešenia vypíšete dve čísla – minimálny počet paličiek ktoré treba pridať aby sme strom zase spájali (označme si ho e) a najmenší možný súčet kontrastov pridaných paličiek. Následne vypíšete e riadkov popisujúcich jeden spôsob, ako tieto hodnoty dosiahnuť – v každom riadku vypíšete čísla dvoch vrcholov, ktoré chcete spojiť. Medzi riešeniami pre jednotlivé vrcholy vypíšete prázdny riadok

Na získanie bodov za sady 1 a 3 vám stačí len vypísať e , minimálny kontrast a následne e riadkov, pričom v každom riadku môžu byť ľubovoľné dve čísla. Čiže nikto nebude kontrolovať, či vami nájdený návod funguje.

Príklad

vstup

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

výstup

```
1 1
3 4

1 1
3 4

0 0

0 0

1 1
5 6

2 2
4 3
4 5

1 1
6 5

0 0

0 0

0 0
```

Vzorový vstup obsahuje dva stromy, pre ktoré budeme úlohu riešiť. V prvom z nich, keď odstránime vrchol 1, strom sa nám rozpadne na dve časti, jedna časť sú vrcholy 2 a 3 spojené hranou a druhá časť je samostatný vrchol 4. Pridaním hrany medzi vrcholy 3 a 4 strom opravíme. Kontrast tohto pospájania je rovný 1. Vieme si rozmyslieť, že menší počet hrán či kontrast nevieme dosiahnuť.