

CEOI 2026



2. deň
(slovensky)

Day 2
(Slovak)

Rezanie kvetov

Časový limit: 4 s Limit pamäte: 256 MiB

V komunitnej záhrade CEOI pestujeme zbierku špeciálnych kvetov, ktoré spolu prepleťajú svoje korene. Ak ich prerežeme, znova dorastú, pokiaľ nerežeme príliš agresívne a nespôsobíme nenapraviteľné škody.

Ak dvojica kvetov a a b prepletie spolu svoje korene, povieme, že sú „spojené“. V opačnom prípade sú „nespojené“. Korene rastú podľa nasledujúceho pravidla: Nech a a b sú dva nespojené kvety. Ak existujú aspoň 2 ďalšie kvety c a d také, že každý z kvetov a a b je spojený s oboma c aj d , potom medzi a a b vyrastú korene a stanú sa spojenými.

Tieto kvety už rastú nejaký čas a všetky korene, ktoré mohli vyrásť podľa vyššie uvedeného pravidla, už vyrástli. Inými slovami, ak sú dva kvety a a b oba spojené s niektorými kvetmi c a d , potom je zaručené, že a a b sú spojené aj navzájom.

Potrebuje túto záhradu vykopať a presunúť na ďalšie miesto konania CEOI. Aby sme zjednodušili presun, chceli by sme prerezať čo najviac koreňov. Chceme však, aby kvety znova dorástli do svojho súčasného stavu. Aký je maximálny počet spojených dvojíc kvetov, ktorých spojenia môžeme prerezať tak, aby aj napriek tomu znova dorástli do svojho súčasného stavu? Nezáleží na tom, koľko iterácií rastu by to vyžadovalo.

Vstup

Prvý riadok vstupu obsahuje dve celé čísla n a m oddelené medzerou, počet kvetov a počet existujúcich spojení medzi nimi. Nasleduje m riadkov, z ktorých každý obsahuje dvojicu celých čísel a_i a b_i , označujúcu, že kvety a_i a b_i sú spojené. Kveti sú označené celými číslami $1 \dots n$. Je zaručené, že vstup spĺňa pravidlo opísané v zadání úlohy.

Obmedzenia

- $1 \leq n \leq 1000$
- $1 \leq m \leq 10^5$

Podúlohy

- Podúloha 1 (20 bodov): $n \leq 10$ a $m \leq 20$
- Podúloha 2 (14 bodov): $m = \frac{n \cdot (n-1)}{2}$
- Podúloha 3 (15 bodov): Je zaručené, že každý kvet je spojený s najviac 7 ďalšími kvetmi.
- Podúloha 4 (15 points): $n \leq 50$ and $m \leq 1000$
- Podúloha 5 (36 bodov): Žiadne ďalšie obmedzenia.

Výstup

Vypíšte jedno celé číslo — maximálny počet spojení, ktoré môžeme prerezať.

Príklad

Vstup

```

9 14
1 2
1 4
1 5
2 4
2 5
3 4
4 5
3 6
4 6
6 7
6 9
7 9
8 9
5 8

```

Výstup

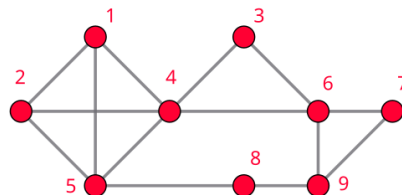
```

2

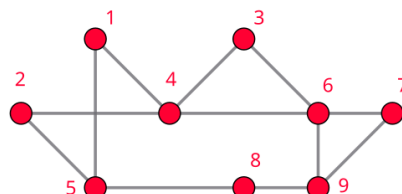
```

Komentár

Kvety v príklade pred akýmkoľvek rezaním. Dá sa overiť, že na základe opísaného postupu rastu nemôžu vzniknúť žiadne ďalšie spojenia.



Kvety v príklade po prerezaní majú o 2 spojenia menej. Spojenie medzi 1 a 2 môže znova dorásť, pretože kvety 1 a 2 sú oba spojené s kvetmi 4 a 5. Podobne môže znova dorásť spojenie medzi 4 a 5, pretože kvety 4 a 5 sú oba spojené s kvetmi 1 a 2.



Veže

Časový limit: 1 s Limit pamäte: 256 MiB

Máte n počítačov a m veží rozmiestnených na rôznych pozíciách na jednej priamke. Potrebujete spárovať počítače pomocou káblov takým spôsobom, aby každý kábel začínal na nejakom počítači, navštívil niektoré veže a skončil na inom počítači. Kábel môže navštíviť akékoľvek veže (nie len tie medzi tými počítačmi) v ľubovoľnom poradí. Môže veže obchádzať, teda prejsť okolo bez návštevy. Môže tiež nenavštíviť žiadnu vežu a spojiť oba počítače priamo, ale nemôže spojiť počítač so samým sebou. Počet počítačov je párny.

Nech a a b sú pozície dvoch počítačov a nech $x_1, \dots, x_k$ sú pozície veží, ktoré kábel navštívi. Dĺžka kábla je $|a - x_1| + |x_1 - x_2| + \dots + |x_{k-1} - x_k| + |x_k - b|$. Pre daný kábel definujeme jeho skóre ako $f \cdot u - l$, kde l je jeho dĺžka, f je nejaká pevne daná konštanta a u je počet jedinečných veží, ktoré kábel navštívil spomedzi $x_1, \dots, x_k$. Viaceré káble môžu navštíviť tú istú vežu a táto veža prispieva do skóre každého z týchto káblov.

Vašou úlohou je vypočítať maximálny možný súčet skóre káblov ak spárujete všetky počítače (t.j. každý počítač musí patriť presne do jednej dvojice, alebo inak povedané, musí byť prepojený presne jedným káblom).

Vstup

Prvý riadok obsahuje počet testovacích prípadov T . Ďalej zaradom nasledujú testovacie prípady. Každý testovací prípad pozostáva z troch riadkov. Prvý riadok obsahuje tri celé čísla n , m a f — počet počítačov, počet veží a konstantu f . Druhý riadok obsahuje n celých čísel $a_1, a_2, \dots, a_n$ — pozície počítačov. Tretí riadok obsahuje m celých čísel $b_1, b_2, \dots, b_m$ — pozície veží.

Obmedzenia

Nech N a M sú súčty hodnôt n a m vo všetkých testovacích prípadoch.

- $1 \leq T \leq 10^4$
- $1 \leq N, M \leq 2 \cdot 10^5$
- $0 \leq f \leq 10^9$
- n je párne.
- $1 \leq a_i, b_i \leq 10^9$
- Všetky pozície počítačov a veží sú navzájom odlišné (v rámci jedného testovacieho prípadu).

Podúlohy

- Podúloha 1 (5 bodov): $N \leq 5000$, $m = 1$
- Podúloha 2 (10 bodov): $T \leq 20$, $n \leq 10$, $m \leq 100$
- Podúloha 3 (27 bodov): $N, M \leq 5000$
- Podúloha 4 (21 bodov): $N \leq 5000$
- Podúloha 5 (37 bodov): Bez ďalších obmedzení.

Výstup

Vypíšte T celých čísel, každé na vlastnom riadku — maximálny možný súčet skóre káblov pre každý testovací prípad.

Príklad

Vstup

```
4
2 1 100
1 10
11
4 1 10
2 4 6 8
20
4 1 10
2 4 6 8
5
6 3 10
2 13 4 8 6 10
5 1 9
```

Výstup

```
89
-4
12
51
```

Vim

Časový limit: 1 s Limit pamäte: 256 MiB

Ťažko by bolo nájsť niečo viac *geekovské* ako starý Vim — editor, v ktorom dokážeme robiť veci, o ktorých sa nám v „bežných“ editoroch môže len snívať. Teda ak sme ochotní investovať týždeň do učenia a mesiac do cvičenia, aby sa nezvyčajné, ale mocné príkazy, ktoré ponúka, stali našou „svalovou pamäťou“. Nuž, rovnako ako v iných editoroch, aj vo Vime máme *kurzor*, ktorý sa vždy nachádza na nejakom znaku. Na začiatku sa nachádza na prvom znaku textu. A rovnako ako v iných editoroch, aj vo Vime existuje *schránka* (clipboard), ktorá slúži na uchovávanie (a nepriamo aj kopírovanie a presúvanie) častí textu. Schránka je na začiatku prázdna.

V tejto úlohe budeme predpokladať, že sa v editore na začiatku nachádza presne jeden znak '-' (mínus), a naším cieľom je vytvoriť presne n po sebe idúcich znakov mínus. Pomôžu nám pri tom štyri príkazy:

- **h**: Ak sa kurzor nachádza na prvom znaku, tento príkaz neurobí nič, inak posunie kurzor na vedľajší znak doľava.
- **l**: Ak sa kurzor nachádza na poslednom znaku, tento príkaz neurobí nič, inak posunie kurzor na vedľajší znak doprava.
- **Y**: Postupnosť znakov od pozície kurzora až po koniec textu sa skopíruje do schránky, pričom „prepíše“ jej predchádzajúci obsah. (Ak sa kurzor nachádza na prvom znaku textu, do schránky sa skopíruje celý text.)
- **P**: Kópia textu uloženého v schránke sa vloží pred znak, na ktorom sa nachádza kurzor, a kurzor sa presunie na posledný vložený znak. Obsah schránky sa nemení. Ak je schránka prázdna, nič sa nestane.

Napište program, ktorý načíta číslo n a vypíše minimálny počet príkazov, čo vo Vime treba, aby ste skončili s presne n po sebe idúcimi znakmi '-'. Program má tiež vypísať príklad postupnosti príkazov s minimálnym počtom príkazov.

Vstup

Prvý riadok obsahuje počet testovacích prípadov t . Nasledujúcich t riadkov obsahuje testovacie prípady s požadovanou dĺžkou reťazca n .

Obmedzenia

- $1 \leq t \leq 100$
- $1 \leq n \leq 10^7$

Podúlohy

Ak vypíšete iba správny počet príkazov, ale nesprávny alebo chýbajúci príklad postupnosti príkazov, tak získate polovicu bodov za danú podúlohu.

- Podúloha 1 (20 bodov): $n \leq 100$

- Podúloha 2 (8 bodov): $n \leq 1000$
- Podúloha 3 (18 bodov): $n \leq 10^4$
- Podúloha 4 (18 bodov): $n \leq 10^5$
- Podúloha 5 (18 bodov): $n \leq 10^6$
- Podúloha 6 (18 bodov): Bez ďalších obmedzení.

Výstup

Pre každý testovací prípad vypíšte na samostatný riadok požadovaný minimálny počet príkazov a príklad takejto postupnosti príkazov.

Príklad

Vstup

2
21
2

Výstup

10 YPYPhPYPPP
2 YP

Komentár

Ako môžeme vidieť v nasledujúcej tabuľke, v prvom prípade dáva postupnosť YPYPhPYPPP správny výsledok. Znak '=' v stĺpci Obrázovka predstavuje znak '-', na ktorom sa nachádza kurzor.

Krok	Príkaz	Obrázovka	Schránka
0		=	(prázdna)
1	Y	=	-
2	P	=-	-
3	Y	=-	--
4	P	-----	--
5	h	-----	--
6	P	-----	--
7	Y	-----	-----
8	P	-----=-----	-----
9	P	-----=-----	-----
10	P	-----=-----	-----