

CEOI 2026



2. den
(česky)

Day 2
(Czech)

Prořezání květin

Časový limit: 4 s Limit paměti: 256 MiB

V komunitní zahradě CEOI pěstujeme zvláštní květiny, které si navzájem proplétají svoje kořeny. Pokud jejich kořeny přeřízneme, tak dorostou za předpokladu, že jsme řezali opatrně a vyhnuli jsme se trvalým škodám.

Pokud dvojice květin a a b mají navzájem spletené kořeny, budeme jim říkat "propojené". Jinak jim budeme říkat "rozpojené". Kořeny rostou podle následujícího pravidla: Nechť a a b jsou rozpojené květiny. Pokud existují 2 jiné květiny c a d a každá z květin a , b je propojená s oběma květinami c i d , tak kořeny mezi a a b dorostou a dvojice a , b se stane se propojenou.

Naše květiny již chvíli rostly, a tak všechny kořeny, které mohli dorůst podle pravidla výše, již dorostly. Jinými slovy, pokud dvě květiny a a b jsou obě propojené s nějakými jinými květinami c a d , pak je zaručeno, že a a b jsou propojené.

Nyní potřebujeme celou zahradu vykořenit a přemístit na příští CEOI. Abychom si zjednodušili práci, rádi bychom přeřízli co nejvíce kořenů. Nicméně chceme, aby květiny dorostly zpátky do původního stavu. Jaký je největší počet propojených páru, které můžeme rozříznout? Nezáleží na tom, kolik iterací by růst květin trval.

Vstup

První řádek vstupu obsahuje dvě mezerou oddělená celá čísla n a m , počet květin a počet propojených dvojic. Následuje m řádků, každý obsahující dvojici celých čísel a_i a b_i udávající, že květiny a_i a b_i jsou propojené. Květiny jsou označeny celým čísly od 1 do n . Je zaručeno, že vstup splňuje pravidlo v zadání.

Omezení

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

Podúlohy

- Podúloha 1 (20 bodů): $n \leq 10$ a $m \leq 20$
- Podúloha 2 (14 bodů): $m = \frac{n \cdot (n-1)}{2}$
- Podúloha 3 (15 bodů): Každá květina je propojena s nejvýše 7 dalšími květinami.
- Podúloha 4 (15 bodů): $n \leq 50$ a $m \leq 1000$
- Podúloha 5 (36 bodů): Bez dalších omezení.

Výstup

Vypište jedno celé číslo – maximální počet propojení, které můžeme přeříznout.

Příklad

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

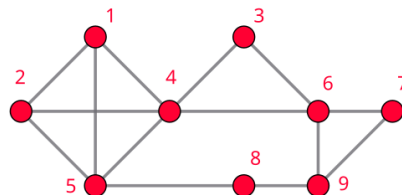
```

2

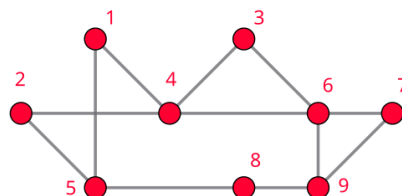
```

Vysvětlení

Květiny v příkladu před jakýmkoliv řezáním. Můžeme ověřit, že žádné další propojení nemůže vyrůst podle pravidla v zadání.



Květiny v příkladu po řezání mají o 2 spojení méně. Spojení mezi 1 a 2 může vyrůst, protože květiny 1 a 2 jsou obě propojené s 4 i 5. Obdobně, spojení mezi 4 a 5 může vyrůst, protože 4 a 5 jsou obě propojené s 1 a 2.



Věže

Časový limit: 1 s Limit paměti: 256 MiB

Máme n počítačů a m věží, všechny umístěné na různých pozicích na přímce. Vaším úkolem je propojit páry počítačů kabely, tak, aby každý kabel začal u nějakého počítače, navštívil nějaké věže a skončil u jiného počítače. Kabel může navštívit věže v libovolném pořadí. Může také přeskočit věže skrze které prochází. Rovněž může nenavštívit žádné věže a propojit dva počítače napřímo. Ale nemůže propojit počítač sám se sebou. Slibujeme, že počet počítačů je sudý.

Nechť a a b jsou pozice dvou počítačů a nechtě $x_1, \dots, x_k$ jsou pozice věží, které kabel navštíví. Délka kabelu pak je $|a - x_1| + |x_1 - x_2| + \dots + |x_{k-1} - x_k| + |x_k - b|$. Pro kabel definujeme jeho skóre jako $f \cdot u - l$, kde l je délka kabelu, f je pevná konstanta a u je počet unikátních věží, které kabel navštíví (v rámci $x_1, \dots, x_k$). Více kabelů může navštívit stejnou věž a ta pak přispívá do skóre každého z těchto kabelů.

Vaším úkolem je spočítat maximální možný součet skóre kabelů spojujících všechny počítače do párů (tedy každý počítač musí patřit do právě jednoho páru, neboli být připojen právě jedním kabelem).

Vstup

První řádek obsahuje počet problémů, T . Problémy následují jeden za druhým. Každý problém se skládá ze tří řádků. První řádek obsahuje tři čísla n , m a f – počet počítačů, počet věží a konstantu f . Druhý řádek obsahuje n celých čísel $a_1, a_2, \dots, a_n$ – pozice počítačů. Třetí řádek obsahuje m celých čísel $b_1, b_2, \dots, b_m$ – pozice věží.

Omezení

Nechť N je součet n přes všechny problémy a obdobně M je součet m přes všechny problémy.

- $1 \leq T \leq 10^4$
- $1 \leq N, M \leq 2 \cdot 10^5$
- $0 \leq f \leq 10^9$
- n je sudé.
- $1 \leq a_i, b_i \leq 10^9$
- Všechny pozice počítačů a věží jsou unikátní (v rámci jednoho problému).

Podúlohy

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

Výstup

Vypište T čísel – maximální možný součet skóre kabelů pro každý problém.

Příklad

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 paměti: 256 MiB

Bylo by obtížné najít něco více *geekovského* než je starý dobrý Vim – editor, ve kterém můžeme dělat věci, o kterých si v „normálních“ editorech můžeme nechat jenom zdát. To samozřejmě jenom pokud jsme ochotni strávit týden učením se a měsíc procvičováním, aby se neobvyklé, ale mocné příkazy, které nabízí, staly součástí naší „svalové paměti“. Stejně jako v ostatních editorech, máme i ve Vimu *kurzor*, který se vždy nachází na některém znaku. Na začátku je na prvním znaku textu. Podobně je ve Vimu také *schránka*, která slouží k ukládání (a nepřímo ke kopírování a přesouvání) kousků textu. Schránka je na začátku prázdná.

Pro tuto úlohu budeme předpokládat, že se v editoru na začátku nachází právě jeden znak '-' (mínus) a naším cílem je skončit s přesně n po sobě jdoucích mínusů. K tomu budeme používat následující čtyři příkazy:

- **h**: Pokud je kurzor na prvním znaku textu, tento příkaz nic nedělá. Jinak posune kurzor o jeden znak doleva.
- **l**: Pokud je kurzor na posledním znaku textu, tento příkaz nic nedělá. Jinak posune kurzor o jeden znak doprava.
- **Y**: Posloupnost znaků od pozice kurzoru až do konce textu se zkopíruje do schránky a „přepíše“ její předchozí obsah. (Pokud je kurzor na prvním znaku, zkopíruje se celý text.)
- **P**: Kopie textu uloženého ve schránce se vloží před znak, na kterém se nachází kurzor. Kurzor se přesune na poslední vložený znak. Obsah schránky se nezmění. Pokud je schránka prázdná, nic se nestane.

Napište program, který načte číslo n a vypíše minimální počet příkazů potřebných ke skončení s přesně n po sobě jdoucími znaky '-' ve Vimu za popsanych podmínek. Vypíšte rovněž takovou posloupnost příkazů s minimální délkou.

Vstup

První řádek obsahuje počet problémů t , které musíte vyřešit. Pro každý problém dostanete na samostatném řádku číslo n popisující požadovanou délku konečného textu.

Omezení

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

Podúlohy

Pokud vypíšete pouze správný počet příkazů, ale nevypíšete posloupnost příkazů, nebo ji vypíšete špatně, dostanete polovinu bodů za danou podúlohu.

- Podúloha 1 (20 bodů): $n \leq 100$

- Podúloha 2 (8 bodů): $n \leq 1000$
- Podúloha 3 (18 bodů): $n \leq 10^4$
- Podúloha 4 (18 bodů): $n \leq 10^5$
- Podúloha 5 (18 bodů): $n \leq 10^6$
- Podúloha 6 (18 bodů): Bez dalších omezení.

Výstup

Pro každý problém vypište minimální počet příkazů a libovolnou takto dlouhou posloupnost příkazů, která skončí s n znaky.

Příklad

Vstup

2
21
2

Výstup

10 YPYPhPYPPP
2 YP

Komentář

V následující tabulce můžeme vidět že v prvním problému posloupnost YPYPhPYPPP dává správný výsledek. Znak = ve sloupci Obrazovka představuje znak '-', na kterém se nachází kurzor.

Krok	Příkaz	Obrazovka	Schránka
0		=	(prázdná)
1	Y	=	-
2	P	=-	-
3	Y	=-	--
4	P	=---	--
5	h	=---	--
6	P	=----	--
7	Y	=----	-----
8	P	-----=-----	-----
9	P	-----=-----	-----
10	P	-----=-----	-----