

Zadání úloh soutěže Kasiopea 2013

1 Skořápký

Uvažme ryze hypotetického skořápkáře, který každou hru provozuje férově.

Takovýto skořápkář má tři kalíšky, pod jeden z nich před Vámi umístí kuličku. Pak skořápkář N -krát opakuje následující operaci: Vybere dva kalíšky a rychlým šoupavým pohybem zamění pozici těchto dvou kalíšků.

Určete konečnou pozici kuličky.

První řádek souboru *vstup.in* obsahuje číslo N a číslo k označující výchozí pozici kalíšku s kuličkou. Následuje N řádků, na každém z nich je dvojice čísel p a q , což znamená, že skořápkář zamění kalíšky na těchto pozicích.

Čísla k , p a q jsou z množiny $\{1, 2, 3\}$.

Pro prvních 5 vstupů bude $N \leq 100$, pro zbývajících $N \leq 100\,000$.

Na jediný řádek souboru *vystup.out* vypište jediné číslo – pozici kuličky.

Ukázkový vstup:

```
5 2
1 2
3 2
3 1
1 2
2 1
```

Ukázkový výstup: 3

2 Permutace

*Tradiční soutěž otužilců
probíhala už několik dní
ve vodě zbyli blázni
sportovci jsou zmotivovaní
ona šla po mostu
a tam si sundala
tu svoji uniformu momentálně...*

(Vypsána $f_i X_a$)

V této fázi soutěže je tradičním zvykem si s sebou přivést co nejzajímavější horký nápoj a podělit se o něj.

Protože otužilec nedá svůj nápoj z ruky, pokud nedostane náhradou jiný, byl navržen následující systém: Každý z N otužilců má přidělené unikátní číslo z rozsahu od 0 do $N - 1$. Dále má každý otužilec předepsáno, kterému otužilci předává svůj nápoj. (To může být i on sám, pokud nechce nápoj sdílet, je nemocný nebo podobně.) Je zaručeno, že každý otužilec od někoho nápoj obdrží.

V pravidelných intervalech dochází k hromadnému předávání nápojů, otužilec předá nápoj předepsanému následníkovi a obdrží nápoj od jiného otužilce.

Vášim úkolem je pro zadanou předávací tabulku určit, kolik nejméně je nutno vyhlásit hromadných předávání k tomu, aby každý otužilec měl zpět svůj nápoj. (Musí být vyhlášeno alespoň jedno předávání.)

První řádek souboru *vstup.in* obsahuje počet otužilců N . Druhý řádek skýtá N mezerou oddělených hodnot: i -tá hodnota udává, komu předává svůj nápoj i -tý otužilec.

Na jediný řádek souboru *vystup.out* vypište minimální potřebný počet hromadných předávání.

V prvních 5 vstupech je $N \leq 30$, ve zbývajících je $N \leq 10^4$. Výstup se vleze do 32-bitového integeru.

Ukázkové vstupy:

```
vstup.in
4
3 2 1 0
```

```
vystup.out
2
```

(Při prvním předávání si vymění nápoje dvojice 0,1 a dvojice 2,3. Po druhém předávání se vrátí nápoje majitelům.)

```
vstup.in
8
2 1 4 5 0 6 7 3
```

```
vystup.out
12
```

3 Barevné proužky

Emil dostal k narozeninám opravdu široký linkovaný papír. Rozhodl se do každého řádku na nějaké místo nakreslit barevný proužek. Svůj výtvar pak přinesl ukázat mamince, ta se na celé dílo dlouze podívala, pochválila Emila, jaký je to malý umělec, a zeptala se: „Ve kterém sloupečku jsi použil nejvíce barev?“

Proužek je daný dvojicí celočíselných indexů (začátku Z a konce K). Nakreslení proužku si můžeme představit tak, že zasahuje do všech jednotkových sloupečků mezi Z a K včetně.

Na prvním řádku souboru *vstup.in* dostanete číslo N udávající počet řádků na Emilově papíře. Na následujících N řádcích bude na každém trojici čísel udávající index začátku, index konce a barvu proužku.

Omezení:

$$1 \leq N \leq 300\,000$$

- $1 \leq \text{hodnoty barev} \leq N$
- $1 \leq \text{index začátku} \leq \text{index konce} \leq 10^9$
- 6 bodů je možné získat pro velikosti intervalů menší než 2500

Na jediný řádek souboru *vystup.out* vypište dvě čísla oddělená mezerou. Maximální počet různých barev a index sloupečku, ve kterém je můžeme najít. Pokud existuje více možných indexů, odpovězte ten nejmenší z nich.

Ukázkový vstup:

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

Ukázkový výstup:

```
3 3
```

4 Nulový strom

Strom je množina N vrcholů a $N - 1$ hran spojujících vrcholy tak, že mezi každými dvěma vrcholy vede právě jedna cesta.

Podstrom daného stromu je strom, jehož vrcholy a hrany byly obsaženy v původním stromu. (Podstromem tedy může být i celý původní strom, stejně jako jeden osamocený vrchol.)

Máte zadaný strom o N vrcholech očíslovaných 0 až $N - 1$. Každý vrchol i má celočíselné ohodnocení v_i .

Jeden tah sestává ze dvou kroků:

- 1) Vybrat podstrom obsahující vrchol 0.
- 2) Ohodnocení všech vybraných vrcholů zvýšit nebo snížit o 1.

Určete nejmenší počet tahů potřebných k tomu, aby všechny vrcholy stromu měly ohodnocení 0.

Na prvním řádku souboru *vstup.in* se vyskytuje číslo N . Na druhém řádku hodnoty v_0 až v_{N-1} . Na každém ze zbývajících $N - 1$ řádků se nachází dvojice čísel u a w značící, že vrcholy u a w jsou spojeny hranou.

Pro prvních 5 vstupů bude $N \leq 100$ a $|v_i| \leq 100$. Ve zbývajících vstupech platí $N \leq 10^5$ a $|v_i| \leq 10^6$.

Na jediný řádek souboru *vystup.out* vypište požadovaný minimální počet tahů.

Ukázkové vstupy:

```
vstup.in
3
1 -1 1
0 1
0 2
```

```
vystup.out
3
```

```
vstup.in
5
0 2 1 4 3
1 2
3 4
1 4
0 2
```

```
vystup.out
8
```

5 Průchod přes dvě patra

Eda pracuje na zámku jako zámečnick a průvodce. Tam nyní otvírají novou dvoupatrovou chodbu, ve které v obou patrech je L zavřených dveří. Každé dveře na sobě mají nějaký typ zámku. Mezi každými dvěma dveřmi je možné přejít po schodech mezi patry.

Aby prošel chodbou, dostal Eda obrovský svazek N klíčů. Naneštěstí vždy může vzít pouze první klíč v řadě a ten buď jednou použít a zahodit, anebo jej zahodit rovnou. Ví ale, jaké typy klíčů a v jakém pořadí ze svazku bude dostávat.

Aby jeho návštěvníci byli co nejvíce spokojeni, chce při cestě s nimi otvírat dveře tak, aby bylo potřeba co nejméněkrát změnit patro, přičemž začít můžou v kterémkoliv z nich. Vaším úkolem je pro dané typy dveří a posloupnost klíčů zjistit, kolikrát nejméně mezi patry budou muset přejít.

Na prvním řádku souboru *vstup.in* dostanete čísla N a L . N je počet klíčů ve svazku a L je počet dveří v chodbě (všechny dvojice dveří jsou přesně nad sebou). Na druhém a třetím řádku je vždy L čísel, která popisují typy dveří v jednotlivých patrech. Na čtvrtém, posledním, řádku je pak N čísel udávajících posloupnost klíčů, které Eda bude dostávat.

Omezení:

$$\begin{aligned} 1 &\leq N \leq 10\,000 \\ 1 &\leq L \leq 1\,000 \\ 1 &\leq \text{typů klíčů/dveří} \leq 10^9 \end{aligned}$$

Do souboru *vystup.out* vypište jedno číslo udávající minimální počet změn patra. V případě, že chodbu vůbec nelze projít, vypište -1 .

Ukázkový vstup:

```
8 4
1 3 2 1
2 1 4 3
1 2 4 1 1 2 2 3
```

Ukázkový výstup: 2

Chodbu ze vzorového vstupu Eda projde následujícím způsobem:

- 1 zahodí
- 2 použije ve spodním patře
- 4 zahodí
- 1 použije
- 1 zahodí
- přejde do horního patra a použije 2
- 2 zahodí
- přejde do spodního patra a použije 3

Celkem provede 2 přechody mezi patry a na méně přechodů tento vstup řešit nelze.

6 Obchod s grafy

V dnešní době se ve městě dá koupit téměř cokoliv. Od oblečení a bot, přes potraviny, alkohol, cigarety, až po různé služby jako je kadeřnictví, manikura a tak podobně. Ctirad se rozhodl, že tomu všemu bude dělat konkurenci a že bude prodávat podgrafy.

Ze železných tyček tedy postavil neorientovaný graf. A jelikož se jich k sobě tolik nevejde, tak z každého vrcholu jich vychází maximálně 111. A každá železná tyčka má nějakou svou váhu.

Hned jak dostavil graf, začal podnikat. Po chvíli přichází první, trochu tajemný, zákazník a má na Ctirada opravdu zvláštní požadavek. Říká: „Chtěl bych si koupit tu nejtěžší kružnici se čtyřmi hranami, kterou v tomto grafu máš.“ Pomozte Ctiradovi najít, která to je.

První řádek souboru *vstup.in* obsahuje přirozená čísla N a M , počet vrcholů $4 \leq N \leq 10\,000$ a počet hran $4 \leq M \leq 1\,000\,000$ grafu. Na následujících M řádcích jsou popsány jednotlivé hrany. Na i -tém řádku jsou přirozená čísla x_i , y_i a w_i , kde $1 \leq x_i \leq N$ a $1 \leq y_i \leq N$ jsou čísla vrcholů, které spojuje i -tá hrana, a $1 \leq w_i \leq 256$ je váha i -té hrany. Žádný vrchol se nevyskytuje ve více než 111 hranách.

Na první řádek souboru *vystup.out* vypište váhu nejtěžší kružnice dlouhé čtyři. Na druhý řádek vypište pět čísel vrcholů v pořadí, v jakém jsou spojovány hranami. První a poslední čísla musí být stejná (jedná se totiž o stejný vrchol). Pokud takových kružnic existuje více, vypište libovolnou z nich.

Pokud žádná taková kružnice neexistuje, na výstup vypište jeden řádek se slovem „NEEXISTUJE“.

Příklady:

```
vstup.in
6 9
1 2 10
2 5 11
3 1 10
```

```
6 3 7
1 4 3
2 6 15
5 3 10
4 5 5
4 6 9
```

```
vystup.out
43
2 6 3 5 2
```

```
vstup.in
7 9
1 2 1
2 3 1
1 3 1
3 4 1
3 5 1
5 4 1
5 6 1
6 7 1
2 7 1
```

```
vystup.out
NEEXISTUJE
```