

Zábavný park Magic City

Starosta Taškentu túži po novom dizajne pre zábavný park Magic City. Zadal vám kladné celé číslo K a nasledujúcu sadu požiadaviek:

- Zvoľte nezáporné celé číslo N určujúce počet atrakcií. Tieto atrakcie číslujeme od 0 po $N - 1$.
- Pre každé i (od 0 po $N - 1$ vrátane) priradíte atrakcii i jej typ, ten si označíme $T[i]$. Existuje $2K$ typov atrakcií, jednotlivé typy sú očíslované od 0 po $2K - 1$. Každý typ musí byť priradený aspoň jednej atrakcii. Rôzne atrakcie môžu mať priradený ten istý typ.
- Postavte obojsmerné chodníky. Každý chodník musí spájať dve **rôzne** atrakcie. Medzi dvoma atrakciami však môže viesť viacero chodníkov. Zábavný park **nemusí byť súvislý**, môžu existovať nejaké dvojice atrakcií, medzi ktorými sa nedá prejsť po chodníkoch.

Prieskum trhu nám ukázal, že platí:

1. Každý návštevník si chce pozrieť presne tri (nie nutne rôzne) atrakcie.
2. Žiadny návštevník nechce navštíviť bezprostredne po sebe dve atrakcie s rovnakým typom.
3. Návštevníkom sa nepáči, keď od atrakcie vedie veľa chodníkov.

Starosta chce, aby **všetci** potenciálni návštevníci boli zaručene spokojní. Toto sa dosiahne tak, že váš dizajn dostane ešte dve nové podmienky. Tie si definujeme nižšie.

Nech (t_1, t_2, t_3) je usporiadaná trojica **typov**, teda platí $0 \leq t_1, t_2, t_3 < 2K$. Takúto trojicu voláme **zaujímavá** ak $t_1 \neq t_2$ a zároveň $t_2 \neq t_3$. Všimnite si, že existuje presne $2K \cdot (2K - 1)^2$ zaujímavých trojíc typov.

Podmienka 1: Pre každú zaujímavú trojicu (t_1, t_2, t_3) musí existovať trojica atrakcií a_1, a_2, a_3 (čísel spájajúcich $0 \leq a_1, a_2, a_3 < N$) taká, že:

- Typy atrakcií a_1, a_2, a_3 sú t_1, t_2, t_3 . Formálnejšie: $T[a_1] = t_1, T[a_2] = t_2$ a $T[a_3] = t_3$.
- Medzi a_1 a a_2 je postavený chodník.
- Medzi a_2 a a_3 je postavený chodník.

Je nám jedno, či existuje chodník medzi a_1 a a_3 . Všimnite si tiež, že ak $t_1 = t_3$, tak je dovolené, aby platilo aj $a_1 = a_3$, teda prvá atrakcia môže byť v takom prípade zhodná s tretou.

Podmienka 2: Od každej atrakcie môže viesť **nanajvýš** K chodníkov.

Detaily implementácie

Táto úloha je **output only** a za jednotlivé testy sa dajú získať **častočné body**. Každý test zodpovedá nejakej hodnote K . V každom teste je vašou úlohou nadizajnovat' park, ktorý spĺňa všetky vyššie

uvedené podmienky pre danú hodnotu K . Vaše skóre bude následne závisieť od počtu použitých atrakcií. Menej je lepšie.

Existujú dva spôsoby ako odovzdať riešenie. Pre každý test môžete použiť ľubovoľný z nich. Tieto spôsoby sú:

- **Volanie funkcie**
- **Výstupný súbor**

Ak chcete na odovzdanie riešenia použiť **volanie funkcie**, potrebujete implementovať a odovzdať nasledujúcu funkciu:

```
std::pair<std::vector<int>, std::vector<std::pair<int, int>>> construct(int K)
```

- K : polovica počtu typov atrakcií a tiež maximálny počet chodníkov vedúcich od každej atrakcie
- Táto funkcia bude zavolaná práve raz pre každý test.

Návratovou hodnotou vašej funkcie má byť usporiadaná dvojica (T, E) popisujúca vami nadizajnovaný zábavný park. Nech N je počet atrakcií a M je počet chodníkov v ňom. Potom:

- T je pole dĺžky N udávajúce pre každú atrakciu jej typ.
- E je pole dĺžky M popisujúce jednotlivé chodníky. Pre každé j ($0 \leq j < M$) platí, že $E[j] = (U[j], V[j])$ predstavuje obojsmerný chodník vedúci medzi atrakciami $U[j]$ a $V[j]$.

Ak chcete na odovzdanie riešenia použiť **výstupný súbor**, potrebujete vytvoriť a odovzdať textový súbor v nasledujúcom formáte:

```
N M
T[0] T[1] ... T[N-1]
U[0] V[0]
U[1] V[1]
...
U[M-1] V[M-1]
```

Vaše riešenie je **platné** len vtedy, ak spĺňa nasledovné podmienky:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ pre každé i ($0 \leq i < N$).
- $0 \leq U[j], V[j] < N$ a $U[j] \neq V[j]$ pre každé j ($0 \leq j < M$).
- Spĺňa Podmienky 1 aj 2 zo zadania.

Obmedzenia

- $1 \leq K \leq 50$

Hodnotenie

Je 50 testov (subtaskov): jeden pre každú z hodnôt K od 1 po 50. Presnejšie, testy majú čísla od 1 po 50 a v teste i platí, že premenná K má hodnotu i .

Každý test má priradené svoje skóre S a vysnívanú veľkosť parku P . Tieto hodnoty sú v tabuľke nižšie.

Test	S	P
1	1	2
2	8	12
3	9	24
4	9	40
5	9	50
6 - 10	4	$12 \cdot K$
11 - 12	3	$12 \cdot K$
13 - 50	1	$12 \cdot K$

Pre každý test platí, že ak vaše riešenie nepopisuje platný zábavný park, tak zaň dostanete nulu (a v CMS dostanete verdikt `Output isn't correct`).

Za platný zábavný park sa body počítajú v závislosti od S , P a vášho N nasledovne:

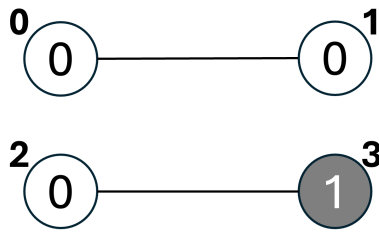
N vzhľadom na P	výsledné body
$N \leq P$	S
$P < N \leq 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \leq 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

Príklad

Uvažujme nasledujúce volanie vašej funkcie:

```
construct(1)
```

V tomto príklade máme $K = 1$, čiže rôznych typov atrakcií je $2K = 2$. Na obrázku nižšie je príklad platného riešenia s $N = 4$ atrakciami a $M = 2$ chodníkmi medzi nimi. Atrakcie 0, 1 a 2 majú všetky typ 0. Posledná atrakcia 3 má typ 1.



Pre $K = 1$ existujú dve zaujímavé trojice typov. Podmienka 1 je splnená, lebo platí:

- Pre trojicu typov $(0, 1, 0)$ si môžeme zvoliť $(a_1, a_2, a_3) = (2, 3, 2)$.
- Pre trojicu typov $(1, 0, 1)$ si môžeme zvoliť $(a_1, a_2, a_3) = (3, 2, 3)$.

Vaša funkcia vie vyššie uvedený zábavný park vrátiť na výstupe tak, že vráti usporiadanú dvojicu $([0, 0, 0, 1], [(0, 1), (2, 3)])$.

Podotkneme ešte, že toto riešenie nemusí byť optimálnym riešením pre $K = 1$.

Ukázkový testovač

Formát vstupu:

K

Formát výstupu:

```
N M
T[0] T[1] ... T[N-1]
U[0] V[0]
U[1] V[1]
...
U[M-1] V[M-1]
```

Všimnite si, že formát výstupu ukázkového testovača zodpovedá požadovanému formátu pre výstupný súbor.