

Čarobno mesto

Župan Taškenta želi prenoviti znameniti zabavišni park Čarobno mesto. Podano je pozitivno celo število K in vaša naloga je zasnovati park na naslednji način:

- Izberite celo število N , število znamenitosti. Znamenitosti so oštevilčene s celimi števili od 0 do $N - 1$.
- Dodajte dvosmerne sprehajalne poti, vsako med parom **različnih** znamenitosti. Obenem je dovoljeno imeti več kot eno sprehajalno pot med istim parom znamenitosti. **Ni nujno, da za vsak par znamenitosti obstaja sprehajalna pot.**
- Vsaki znamenitosti i ($0 \leq i < N$) določite tip $T[i]$. Obstaja $2K$ tipov, oštevilčenih od 0 do $2K - 1$. Vsak tip mora biti dodeljen vsaj eni znamenitosti. Različne znamenitosti so lahko istega tipa.

Raziskovalci tržišča so ugotovili naslednje:

- Vsak obiskovalec najraje obišče tri znamenitosti, pri čemer med obiskom ne želi zaporedno obiskati dveh znamenitosti istega tipa.
- Obiskovalcem niso všeč znamenitosti, ki so stičišče številnih sprehajalnih poti.

Da bi ugodili **vsem** potencialnim obiskovalcem, župan postavi dva pogoja.

Urejen trojček **tipov** (t_1, t_2, t_3) za $0 \leq t_1, t_2, t_3 < 2K$ imenujemo **zanimiv**, če $t_1 \neq t_2$ in $t_2 \neq t_3$. Opomba: t_1 je lahko enak t_3 . Zatorej obstaja $2K \cdot (2K - 1)^2$ zanimivih trojčkov.

1. pogoj: Za vsak zanimiv trojček (t_1, t_2, t_3) mora obstajati trojček znamenitosti a_1, a_2, a_3 (kjer $0 \leq a_1, a_2, a_3 < N$), tako da:

- Se tipi a_1, a_2, a_3 ujemajo z t_1, t_2, t_3 . To pomeni $T[a_1] = t_1$, $T[a_2] = t_2$ in $T[a_3] = t_3$.
- Med a_1 in a_2 obstaja sprehajalna pot.
- Med a_2 in a_3 obstaja sprehajalna pot.

Ni pomembno, ali med a_1 in a_3 obstaja pot. Opomba: Ko je $t_1 = t_3$, znamenitosti a_1 in a_3 lahko predstavljata isto znamenitost.

2. pogoj: Vsaka znamenitost je lahko stičišče največ K sprehajalnih poti.

Naloga sestoji iz 50 podnalog tipa **zgolj-izhod z delnim ocenjevanjem**. Vsaka podnalog je povezana s točno določeno vrednostjo K , vi pa morate zasnovati park, ki za vrednost K zadosti zgornjima pogojema. Vaša ocena je odvisna od števila znamenitosti v vaši rešitvi. Manj kot je znamenitosti, višja je vaša ocena.

Podrobnosti implementacije

Svojo rešitev lahko oddate na dva načina, in sicer:

- **Klic funkcije**
- **Izhodna datoteka**

Za oddajo rešitve preko **klica funkcije** morate implementirati naslednjo funkcijo:

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

- K : polovica števila tipov znamenitosti in hkrati največje število sprehajalnih poti, ki se stikajo v kateri koli znamenitosti.
- Funkcija se kliče natanko enkrat za vsako podnalogo.

Funkcija mora vrniti par (T, E) , ki opisuje zabavišni park. Naj bo M število sprehajalnih poti v vašem parku.

- T : polje dolžine N , ki opisuje tipe znamenitosti.
- E : polje dolžine M , ki opisuje sprehajalne poti. Za vsak $0 \leq j < M$, $E[j] = (U[j], V[j])$ predstavlja dvosmerno sprehajalno pot med znamenitostma $U[j]$ in $V[j]$.

Za oddajo svoje rešitve prek **izhodne datoteke** ustvarite in oddajte besedilno datoteko naslednje oblike:

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

Opomba: Vaša rešitev mora zadostiti naslednjim omejitvam, da bi bila **veljavna**:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ za vsak $0 \leq i < N$ in vsak tip mora biti dodeljen vsaj eni znamenitosti.
- $0 \leq U[j], V[j] < N$ in $U[j] \neq V[j]$ za vsak $0 \leq j < M$.
- Obema zgoraj opisanima pogojevima mora biti zadoščeno.

Omejitve

- $1 \leq K \leq 50$

Ocenjevanje

Obstaja 50 podnalog, po ena za vsako celo število K od 1 do 50. Za i -to podnalogo ($1 \leq i \leq 50$) je vrednost K enaka i .

Vsaka podnaloga ima vrednost S in ciljno število znamenitosti P , kakor je opisano v spodnji tabeli.

Podnaloge	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	$12 \cdot K$

Za vsako podnalogo, če vaša rešitev ne opisuje veljavnega zabaviščnega parka, bodo točke vaše rešitve enake 0 (v CMS je zavedeno kot `Output isn't correct`).

V nasprotnem primeru se vaša ocena izračuna na podlagi N ter parametrov S in P na naslednji način:

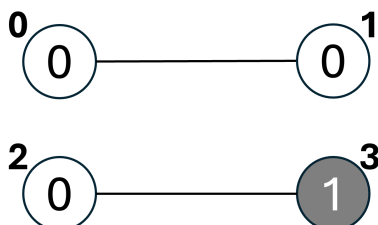
Pogoj	Ocena
$N \leq P$	S
$P < N \leq 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \leq 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

Primer

Razmislite o naslednjem klicu:

```
construct(1)
```

V tem primeru $K = 1$, torej obstajata $2K = 2$ tipa znamenitosti. Spodnja slika prikazuje veljavno rešitev z $N = 4$ znamenitostmi in $M = 2$ potema.



Obstajata zanimiva trojčka:

- Za trojček tipov $(0, 1, 0)$ in $(a_1, a_2, a_3) = (2, 3, 2)$.
- Za trojček tipov $(1, 0, 1)$ in $(a_1, a_2, a_3) = (3, 2, 3)$.

To pomeni, da je 1. pogoj izpolnjen.

Funkcija lahko vrne par $([0, 0, 0, 1], [(0, 1), (2, 3)])$. Opomba: Podana rešitev za ta primer morda ni optimalna za $K = 1$.

Vzorčni ocenjevalnik

Oblika vhoda

```
K
```

Oblika izhoda

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

Opomba: Izhod vzorčnega ocenjevalnika se ujema z zahtevano obliko izhodne datoteke.