

Čarobni grad

Gradonačelnik Taškenta želi redizajnirati poznati zabavni park Magic City. Data vam je pozitivna cijela vrijednost (integer) K i vaš zadatak je da dizajnirate park na sljedeći način:

- Odaberite cijeli broj N , broj atrakcija. Atrakcije su označene (labeled) cijelim brojevima od 0 do $N - 1$.
- Dodajte dvosmjerne pješačke staze, pri čemu svaka povezuje par *različitih* atrakcija. Dozvoljeno je da između istog para atrakcija postoji više od jedne pješačke staze. **Nije potrebno da između svakog para atrakcija postoji mogućnost putovanja pješačkim stazama**
- Za svaki i takav da je $0 \leq i < N$, dodijelite atrakciji i tip $T[i]$. Postoji $2K$ tipova numerisanih od 0 do $2K - 1$. Svaki tip mora (treba) biti dodijeljen barem jednoj atrakciji. Različite atrakcije mogu imati isti tip.

Istraživanje tržišta je otkrilo da:

1. Svaki posjetilac preferira da posjeti tri atrakcije, a da pritom ne posjeti dvije atrakcije iste vrste uzastopno.
2. Posjetioци ne vole atrakcije koje su na kraju od pješačkih staza.

Da bi zadovoljiti **sve** potencijalne posjetioce, gradonačelnik nameće dva uslova vašem dizajnu.

Uređenu trojku **tipova** nazivamo (t_1, t_2, t_3) za $0 \leq t_1, t_2, t_3 < 2K$ **zanimljivo** ako je $t_1 \neq t_2$ i $t_2 \neq t_3$. Imajte na umu da t_1 može biti jednako t_3 . Dakle, postoji $2K \cdot (2K - 1)^2$ zanimljivih trojki.

Uslov 1: Za svaku zanimljivu trojku (t_1, t_2, t_3) , moraju postojati tri atrakcije a_1, a_2, a_3 ($0 \leq a_1, a_2, a_3 < N$) takve da:

- Tipovi a_1, a_2, a_3 odgovaraju t_1, t_2, t_3 . To jest, $T[a_1] = t_1$, $T[a_2] = t_2$ i $T[a_3] = t_3$.
- Između a_1 i a_2 nalazi se pješačka staza.
- Između a_2 i a_3 nalazi se pješačka staza.

Nije bitno da li postoji pješačka staza između a_1 i a_3 . Takođe imajte na umu da kada je $t_1 = t_3$, atrakcije a_1 i a_3 mogu biti iste.

Uslov 2: Svaka atrakcija može biti povezana sa krajnjom tačkom **najviše** K pješačkih staza.

Ovaj zadatak se sastoji od 50 izlaznih (output-only) podzadataka s **parcijalnim bodovanjem**. Svaki podzadatak odgovara određenoj vrijednosti od K , a vi morate dizajnirati park koji zadovoljava sve gore navedene uslove za tu vrijednost od K . Vaš rezultat zavisi od broja atrakcija u vašem rješenju - manji broj atrakcija daje veći ili jednak rezultat.

Detalji implementacije

Postoje dvije metode za slanje rješenja, a za svaki podzadatak možete koristiti bilo koju od njih:

- **Poziv na proceduru** (procedure call)
- **Izlazna datoteka** (output file)

Da biste poslali svoje rješenje putem **poziva procedure**, trebali biste implementirati sljedeću proceduru.

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

- K : polovina broja tipova atrakcija, ujedno i najveći dozvoljeni broj pješačkih staza čija krajnja tačka može biti svaka atrakcija.
- Ova procedura se poziva tačno jednom za svaki podzadatak.

Procedura treba da vrati par (T, E) koji opisuje zabavni park. Neka je M broj pješačkih staza u vašem parku.

- T : niz dužine N koji opisuje tipove atrakcija.
- E : niz dužine M koji opisuje staze. Za svako $0 \leq j < M$, $E[j] = (U[j], V[j])$ predstavlja dvosmjernu stazu između udaljenih atrakcija $U[j]$ i $V[j]$.

Da biste poslali svoje rješenje putem **izlazne datoteke**, kreirajte i pošaljite tekstualnu datoteku u sljedećem formatu:

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

Imajte na umu da vaše rješenje mora zadovoljiti sljedeća ograničenja da bi se smatralo **validnim**:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ za svakih $0 \leq i < N$, svaki tip mora biti dodijeljen najmanje jednoj atrakciji
- $0 \leq U[j], V[j] < N$ i $U[j] \neq V[j]$ za svaki $0 \leq j < M$.
- Uslovi 1 i 2 moraju biti ispunjeni.

Ograničenja

$$1 \leq K \leq 50$$

Bodovanje

Postoji 50 podzadataka, po jedan za svaki cijeli broj K od 1 do 50 . Za podzadatak i ($1 \leq i \leq 50$), vrijednost K je i .

Svaki podzadatak ima rezultat S i ciljan broj atrakcija P prema donjoj tabeli.

Podzadaci	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 svaki podzadatak, ako vaše rješenje ne opisuje valjan zabavni park, tada će rezultat vašeg rješenja biti 0 (prijavljeno kao `Output isn't correct` u CMS-u).

U suprotnom, vaš rezultat se izračunava na osnovu N i parametara S i P na sljedeći način:

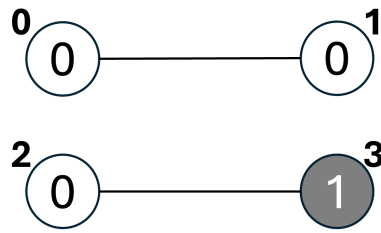
Stanje	Rezultat
$N \leq P$	S
$P < N \leq 2P$	$(0.4 + 0.3 \cdot \frac{2P-N}{P}) \cdot S$
$2P < N \leq 2000$	$(0.1 + 0.3 \cdot \frac{2P}{N}) \cdot S$

Primjer

Razmotrite sljedeći poziv:

```
construct(1)
```

U ovom primjeru, $K = 1$, tako da postoji $2K = 2$ vrsta atrakcija. Donja slika prikazuje valjano rješenje sa $N = 4$ atrakcija i $M = 2$ pješačkih staza. Atrakcije 0, 1, 2 su tipa 0 , a atrakcija 3 je tipa 1 .



Postoje dvije zanimljive trojke:

- Za tip trojke $(0, 1, 0)$, možemo odabrati $(a_1, a_2, a_3) = (2, 3, 2)$.
- Za tip trojke $(1, 0, 1)$, možemo odabrati $(a_1, a_2, a_3) = (3, 2, 3)$.

To znači da je Uslov 1 ispunjen.

Procedura može vratiti par $([0, 0, 0, 1], [(0, 1), (2, 3)])$. Imajte na umu da ponuđeno rješenje za ovaj primjer možda nije optimalno za $K = 1$.

Primjer ocjenjivača

Format unosa:

K

Izlazni format:

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

Imajte na umu da izlaz uzorka ocjenjivača odgovara potrebnom formatu za izlaznu datoteku.