

Magic City

Gradonačelnik Atlante želi redizajnirati poznati zabavni park Magic City. Dat vam je pozitivan cijeli broj K , a vaš zadatak je dizajnirati park na sljedeći način:

- Odaberite cijeli broj N , broj atrakcija. Atrakcije su označene cijelim brojevima od 0 do $N - 1$.
- Dodajte dvosmjerne staze, svaku između para različitih atrakcija. Dozvoljeno je imati više od jedne staze između istog para atrakcija. **Nije obavezno da se pomoću staza može putovati između svakog para atrakcija.**
- 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 treba biti dodijeljen barem jednoj atrakciji. Različite atrakcije mogu imati isti tip.

Istraživanje tržišta pokazalo je da:

1. Svaki posjetilac želi posjetiti tri atrakcije bez uzastopnog posjećivanja dvije atrakcije istog tipa.
2. Posjetioci ne vole atrakcije koje su krajnje tačke velikog broja staza.

Kako bi zadovoljio sve potencijalne posjetioce, gradonačelnik postavlja dva uslova za vaš dizajn.

Uređenu trojku tipova (t_1, t_2, t_3) za $0 \leq t_1, t_2, t_3 < 2K$ nazivamo **interesantnom** ako je $t_1 \neq t_2$ i $t_2 \neq t_3$. Primijetite da t_1 može biti jednak t_3 . Dakle, postoji $2K \cdot (2K - 1)^2$ interesantnih trojki.

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

- Tipovi atrakcija a_1, a_2, a_3 odgovaraju tipovima t_1, t_2, t_3 . To jest, $T[a_1] = t_1$, $T[a_2] = t_2$ i $T[a_3] = t_3$.
- Postoji staza između a_1 i a_2 .
- Postoji staza između a_2 i a_3 .

Nije važno postoji li staza između a_1 i a_3 . Također, kada je $t_1 = t_3$, atrakcije a_1 i a_3 mogu biti iste.

Uslov 2: Svaka atrakcija može biti krajnja tačka najviše K staza.

Ovaj zadatak se sastoji od 50 **output-only** podzadataka sa parcijalnim bodovanjem. Svaki podzadatak odgovara određenoj vrijednosti K , a vi trebate dizajnirati park koji za tu vrijednost K zadovoljava sve navedene uslove. Vaš broj bodova zavisi od broja atrakcija u vašem rješenju: manje atrakcija donosi veći ili jednak broj bodova.

Detalji implementacije

Postoje dva načina za predaju rješenja, a za svaki podzadatak možete koristiti bilo koji od njih:

- **Poziv funkcije**
- **Izlazna datoteka**

Da biste predali rješenje putem **poziva funkcije**, trebate implementirati sljedeću funkciju:

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

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

Funkcija treba vratiti par (T, E) koji opisuje zabavni park. Neka je M broj staza u vašem parku.

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

Da biste predali rješenje putem **izlazne datoteke**, napravite i predajte tekstualnu datoteku u sljedećem formatu:

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

Da bi se vaše rješenje smatralo ispravnim, mora zadovoljiti sljedeća ograničenja:

- $N \leq 2000$
- $0 \leq T[i] < 2K$ za svaki $0 \leq i < N$, a svaki tip treba biti dodijeljen barem 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 zadovoljeni.

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 broj bodova S i ciljni broj atrakcija P prema sljedećoj 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 ispravan zabavni park, broj bodova za vaše rješenje bit će 0 (u CMS-u se prikazuje kao `Output isn't correct`).

U suprotnom, broj bodova računa se na osnovu N i parametara S i P na sljedeći način:

Uslov	Broj bodova
$N \leq P$	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 je $K = 1$, pa postoji $2K = 2$ tipa atrakcija. Slika ispod prikazuje ispravno rješenje sa $N = 4$ atrakcije i $M = 2$ staze. Atrakcije 0, 1 i 2 su tipa 0, a atrakcija 3 je tipa 1.

Postoje dvije interesantne trojke:

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

To znači da je Uslov 1 zadovoljen.

Funkcija može vratiti par $([0, 0, 0, 1], [(0, 1), (2, 3)])$. Primijetite da dato rješenje za ovaj primjer možda nije optimalno za $K = 1$.

Sample Grader

Format ulaza:

K

Format izlaza:

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

Primijetite da izlaz sample gradera odgovara traženom formatu izlazne datoteke.