

Najduži put

Organizatori IOI-a 2023. su u velikom problemu! Gospodin Malnar ukrao je plan za izlet u Ópusztaszer koji je trebao biti sutra. Ali možda još nije prekasno ...

U Ópusztaszer-u se nalazi N znamenitosti označene brojevima od 0 do $N - 1$. Neki parovi tih znamenitosti povezani su *dvosmjernim ulicama*. Svaki par znamenitosti povezan je najviše jednom ulicom. Organizatori *ne znaju* koje znamenitosti su spojene ulicama.

Kažemo da je **gustoća** ulične mreže Ópusztaszra **barem** δ ako svake 3 različite znamenitosti imaju barem δ ulica među sobom. Drugim riječima, ako za svaku trojku znamenitosti (u, v, w) takvu da je $0 \leq u < v < w < N$, među parovima (u, v) , (v, w) i (u, w) barem δ parova povezano je ulicom.

Organizatori *znaju* prirodan broj D takav da je gustoća ulične mreže barem D . Primijetite da vrijednost D ne može biti veća od 3.

Također, organizatori mogu nekoliko puta **nazvati** telefonsku liniju Ópusztaszra kako bi prikupili informacije o uličnoj mreži između određenih znamenitosti. Prilikom jednog poziva, dva neprazna niza znamenitosti $[A[0], \dots, A[P - 1]]$ i $[B[0], \dots, B[R - 1]]$ moraju biti data. Sve znamenitosti koje su date moraju biti međusobno različite, to jest

- $A[i] \neq A[j]$ za svaki i i j takav da je $0 \leq i < j < P$;
- $B[i] \neq B[j]$ za svaki i i j takav da je $0 \leq i < j < R$;
- $A[i] \neq B[j]$ za svaki i i j takav da je $0 \leq i < P$ i $0 \leq j < R$.

Pri svakom pozivu, službenik odgovori postoji li ulica koja povezuje znamenitost iz niza A sa znamenitosti iz niza B . Tačnije, službenik prolazi po svim parovima vrijednost i te j takvim da $0 \leq i < P$ i $0 \leq j < R$. Ako za bilo koji od takvih parova, vrijedi da su znamenitosti $A[i]$ te $B[j]$ povezane ulicom, službenik će vratiti `true`. U suprotnom, službenik će vratiti `false`.

Putovanje dužine l je niz *različitih* znamenitosti $t[0], t[1], \dots, t[l - 1]$, gdje za svako i između 0 i $l - 2$ (uključivo) znamenitost $t[i]$ te znamenitost $t[i + 1]$ su povezane ulicom. Putovanje dužine l naziva se **najdužim putovanjem** ako ne postoji putovanje dužine barem $l + 1$.

Vaš je zadatak pomoći organizatorima pronaći najduži put u Ópusztaszru zivkajući jednog službenika.

Implementacijski detalji

Morate implementirati sljedeće funkcije

```
int[] longest_trip(int N, int D)
```

- N : broj znamenitosti u Ópusztaszru.
- D : garantovana minimalna gustoća ulične mreže.
- Ova funkcija mora vratiti niz $t = [t[0], t[1], \dots, t[l - 1]]$, koji prikazuje neki najduži put.
- Ova funkcija može biti pozivana **više puta** u svakom testnom primjeru.

Gore opisana funkcija može pozivati sljedeću funkciju

```
bool are_connected(int[] A, int[] B)
```

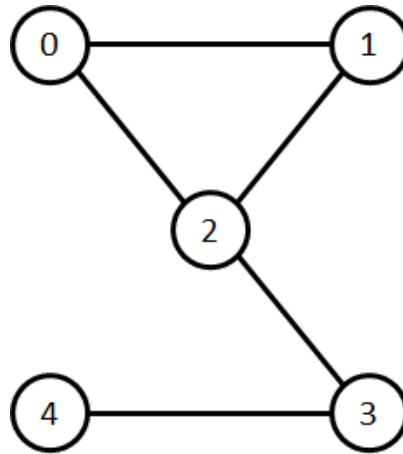
- A : neprazan niz različitih znamenitosti
- B : neprazan niz različitih znamenitosti
- A i B su disjunktni
- Ova funkcija vraća `true` ako postoje znamenitost iz A i znamenitost iz B koje su povezane ulicom. U suprotnom, vraća `false`.
- Ova funkcija smije biti pozvana najviše 32 640 puta unutar svakog poziva funkcije `longest_trip`, i najviše 150 000 puta ukupno.
- Ukupna dužina nizova A i B s kojima je pozivana funkcija tijekom svih poziva funkcije `longest_trip` ne smije biti veća od 1 500 000.

Grader ovog zadatka **nije promjenjiv**. Svako poslano rješenje pokretat će se na istom skupu primjera. Tačnije, vrijednosti N i D , kao i parovi znamenitosti povezanih ulicama, bit će određeni prije svakog poziva funkcije `longest_trip` unutar svakog primjera.

Primjeri

Prvi primjer

Promatrajte slučaj u kojem je $N = 5$, $D = 1$, i ulična mreža je prikazana na skici.



Funkcija `longest_trip` je u tom slučaju pozvana kao:

```
longest_trip(5, 1)
```

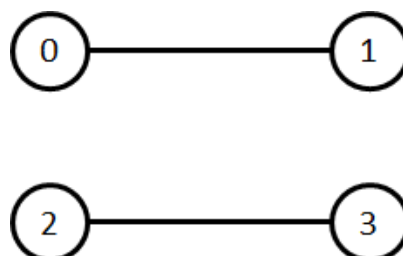
Funkcija poziva `are_connected` na sljedeće načine:

| Poziv | Parovi spojeni ulicom | Vraćena vrijednost |
|---|-----------------------|--------------------|
| <code>are_connected([0], [1, 2, 4, 3])</code> | (0,1) i (0,2) | true |
| <code>are_connected([2], [0])</code> | (2,0) | true |
| <code>are_connected([2], [3])</code> | (2,3) | true |
| <code>are_connected([1, 0], [4, 3])</code> | nema ih | false |

Nakon četvrtog poziva funkcije, *nijedan* od parova (1,4), (0,4), (1,3) i (0,3) nije povezan ulicom. Kako je gustoća mreže barem $D = 1$, iz trojke (0,3,4), vidimo da par (3,4) mora biti povezan ulicom. Slično možemo zaključiti da znamenitosti 0 i 1 moraju biti povezane.

U ovom trenutku, može se zaključiti da je $t = [1, 0, 2, 3, 4]$ put dužine 5 te da ne postoji put dužine veće od 5. Stoga, funkcija `longest_trip` može vratiti `[1, 0, 2, 3, 4]`.

Promatrajte slučaj u kojem je $N = 4$, $D = 1$, i ulična mreža je prikazana na sljedećoj skici.



Funkcija `longest_trip` je u tom slučaju pozvana kao:

```
longest_trip(4, 1)
```

U ovom slučaju, dužina najdužeg puta je 2. Stoga, nakon nekoliko poziva funkcije `are_connected`, funkcija `longest_trip` može vratiti $[0, 1]$, $[1, 0]$, $[2, 3]$ ili $[3, 2]$.

Primjer 2

Podzadatak 0 sadrži dodatan primjer s $N = 256$ znamenitosti. Ovaj primjer je sadržan u datoteci koju možete preuzeti u sistemu takmičenja.

Ograničenja

- $3 \leq N \leq 256$
- Zbir N -ova po svim pozivima funkcije `longest_trip` neće biti veći od 1 024.
- $1 \leq D \leq 3$

Podzadaci

1. (5 bodova) $D = 3$
2. (10 bodova) $D = 2$
3. (25 bodova) $D = 1$. Neka l^* označava duljinu najdužeg puta. Funkcija `longest_trip` ne mora vratiti put duljine l^* . Umjesto toga, mora vratiti put duljine barem $\left\lceil \frac{l^*}{2} \right\rceil$.
4. (60 bodova) $D = 1$

U četvrtom podzadatku, vaš broj bodova biti će određen na temelju broja poziva funkcije `are_connected` tokom jednog pokretanja funkcije `longest_trip`. Neka je q maksimalan broj poziva u svim pozivima funkcije `longest_trip` u svim primjerima podzadatka. Tada vaš broj bodova biti će izračunat na temelju sljedeće tablice:

| Uvjet | Broj bodova |
|---------------------------|-------------|
| $2\,750 < q \leq 32\,640$ | 20 |
| $550 < q \leq 2\,750$ | 30 |
| $400 < q \leq 550$ | 45 |
| $q \leq 400$ | 60 |

Ako, u bilo kojem primjeru, broj poziva funkcije `are_connected` je veći nego što je u ograničenjima opisanim u Implementacijskim detaljima, ili niz koji je vratila funkcija `longest_trip` je netačan, broj bodova vašeg rješenja bit će 0.

Primjer gradera

Neka je C broj scenarija, tačnije, broj poziva funkcije `longest_trip`. Tada probni grader prima ulaz u sljedećem formatu:

- 1. linija : C

Slijede opisi C scenarija.

Probni grader učitava opise svakog slučaja u sljedećem formatu:

- 1. linija : $N D$
- $(1 + i)$ -ta linija ($1 \leq i < N$): $U_i[0] U_i[1] \dots U_i[i - 1]$

Ovdje, svaki U_i ($1 \leq i < N$) je niz dužine i , koji opisuje parove znamenitosti koji su povezani ulicom. Za svaki i te j takav da je $1 \leq i < N$ i $0 \leq j < i$:

- ako su znamenitosti j te i povezane ulicom, tada vrijednost $U_i[j]$ bi trebala biti 1;
- ako nema ulice koja povezuje znamenitosti j te i , tada bi vrijednost trebala biti 0.

U svakom slučaju, prije poziva funkcije `longest_trip`, probni grader provjerava je li gustoća ulične mreže barem D . Ako uvjet nije zadovoljen, ispisuje se poruka `Insufficient Density` te grader prestaje s izvršavanjem.

Ako probni grader primijeti kršenje protokola, izlaz probnog gradera bit će `Protocol Violation: <MSG>`; gdje `<MSG>` označava jednu od sljedećih poruka:

- `invalid array`: u pozivu funkcije `are_connected`, bar jedan od nizova A i B
 - je prazan, ili
 - sadrži član koji nije cijeli broji između 0 i $N - 1$, uključivo, ili
 - sadrži isti član barem dva puta
- `non-disjoint arrays`: u pozivu funkcije `are_connected`, nizovi A i B nisu disjunktni.
- `too many calls`: broj poziva funkcije `are_connected` veći je od 32 640 tokom trenutnog poziva funkcije `longest_trip`, ili veći je ukupno od 150 000.
- `too many elements`: ukupan broj znamenitosti poslan funkciji `are_connected` veći je od 1 500 000.

U suprotnom, neka su članovi niza, koji je vratila funkcija `longest_trip`, redom $t[0], t[1], \dots, t[l - 1]$ za neki nenegativan broj l .

Tada grader ispisuje tri linije, za taj slučaj, u sljedećem formatu:

- 1. linija : l :
- 2. linija : $t[0] t[1] \dots t[l - 1]$
- 3. linija : broj poziva funkcije `are_connected` tokom ovog slučaja

Konačno, probni grader ispisuje:

- $(1 + 3 \cdot C)$ -tu liniju : najveći broj poziva funkcije `are_connected` tokom svih poziva funkcije `longest_trip`