Najdaljši izlet

Organizatorji IOI 2023 so se znašli v težavah. Pozabili so pripraviti jutrišnji izlet v Ópusztaszer. A se še niso predali!

V Ópusztaszerju je N znamenitosti označenih s celimi števili med 0 in $N-1$. Nekaj teh znamenitosti je povezanih z *dvosmernimi* **cestami**. Vsak par znamenitosti je povezan z največ eno cesto. Organizatorji *ne vedo* katere znamenitosti so povezane s cestami.

Gostota cestnega omrežja je **vsaj** δ, če za vsak trojček različnih znamenitosti obstaja vsaj δ cest med njimi. Povedano drugače: Za vse trojčke znamenitosti (u, v, w), kjer $0 \leq u < v < w < N$, je vsaj δ izmed parov (u, v), (v, w) in (u, w) povezanih s cesto.

Organizatorji *poznajo* naravno število D, da je gostota cestnega omrežja najmanj D. Opazimo lahko, da D ne more biti večji od 3.

Organizatorji lahko **pokličejo** v klicni center AMZS v Ópusztaszerju in poizvejo o cestnih povezavah med izbranimi znamenitostmi. Pri vsakem klicu izberejo dve neprazni disjunktni polji znamenitosti $[A[0], \ldots, A[P-1]]$ in $[B[0], \ldots, B[R-1]]$. Znamenitosti morajo biti paroma različne, torej da velja:

- $A[i] \neq A[j]$ za vsak i in j, kjer $0 \leq i < j < P$.
- $B[i] \neq B[j]$ za vsak i in j, kjer $0 \leq i < j < R$.
- $A[i] \neq B[j]$ za vsak i in j, kjer $0 \leq i < P$ in $0 \leq j < R$.

Za vsak tak klic telefonist klicnega centra AMZS sporoči ali obstaja cesta, ki povezuje znamenitost iz A z znamenitostjo v B. Telefonist sporoči `true`, če obstajata i in j, kjer $0 \leq i < P$ in $0 \leq j < R$, in sta $A[i]$ in $B[j]$ povezana s cesto. Sicer sporoči `false`.

Izlet dolžine i je zaporedje *različnih* znamenitosti $t[0], t[1], \ldots, t[l-1]$, kjer sta za vsak i med 0 in vključno $l-2$ znamenitosti $t[i]$ in $t[i+1]$ povezani s cesto. Izlet dolžine l imenujemo **najdaljši izlet**, če ne obstaja izlet dolžine $l+1$.

Pomagaj organizatorjem poiskati najdaljši izlet, s pomočjo klicev v klicni center.

Podrobnosti implementacije

Implementiraj funkcijo:

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

- N: število znamenitosti v Ópusztaszerju.
- D: zagotovljena najmanjša gostota cestnega omrežja.
- Funkcija vrača polje $t = [t[0], t[1], \ldots, t[l-1]]$, ki predstavlja najdaljši izlet.
- Ocenjevalnik lahko pokliče funckijo **večkrat** v istem testnem primeru.

Tvoja funkcija lahko kliče funkcijo:

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

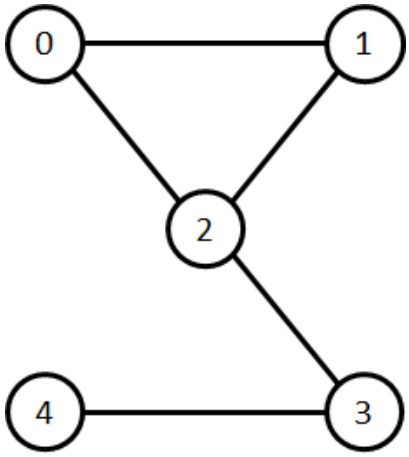
- A: neprazno polje različnih znamenitosti.
- B: neprazno polje različnih znamenitosti.
- A in B morata biti disjunktni.
- Funkcija vrne `true`, če obstaja znamenitost iz A in znamenitost iz B, ki ju povezuje cesta. Drugače vrne `false`.
- Funkcijo lahko pokličeš največ $32\,640$ krat za vsak klic funkcije `longest_trip` in skupno največ $150\,000$ krat.
- Skupna dolžina polj A in B, ki jih podaš tej funkciji z vsemi klici, ne sme presegati $1\,500\,000$.

Ocenjevalnik **ni prilagodljiv**. Vsako oddajo oceni na isti množici testnih primerov. To pomeni, da se N, D in pari znamenitosti, ki so povezani s cesto, ne spreminjajo med klici funkcije `longest_trip` za testni primer.

Primeri

1. primer

Zamislimo si scenarij, kjer $N = 5$, $D = 1$ in ceste povezujejo znamenitosti, kot je prikazano na spodnji sliki.



Ocenjevalnik pokliče funkcijo `longest_trip` na naslednji način:

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

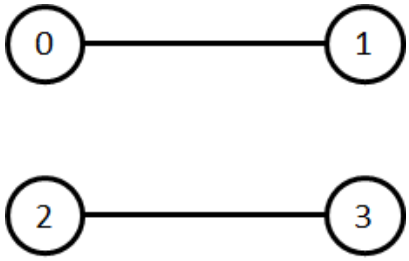
Funkcija lahko kliče `are_connected`, kot je opisano spodaj.

Klic	Pari znamenitosti povezanih s cesto	Vrača
`are_connected([0], [1, 2, 4, 3])`	$(0, 1)$ in $(0, 2)$	`true`
`are_connected([2], [0])`	$(2, 0)$	`true`
`are_connected([2], [3])`	$(2, 3)$	`true`
`are_connected([1, 0], [4, 3])`	ni povezanih znamenitosti	`false`

Po četrtem klicu opazimo, da noben izmed parov $(1, 4)$, $(0, 4)$, $(1, 3)$ in $(0, 3)$ ni povezan s cesto. Ker je gostota omrežja vsaj $D = 1$, lahko vidimo, da je iz trojčka $(0, 3, 4)$ s cesto povezan par $(3, 4)$. Podobno lahko opazimo, da sta znamenitosti 0 in 1 povezani s cesto.

Na tej točki lahko sklenemo, da je izlet $t = [1, 0, 2, 3, 4]$ dolžine 5 in da ne obstaja izlet, ki bi bil daljši. Zatorej lahko funkcija `longest_trip` vrne $[1, 0, 2, 3, 4]$.

Zamislimo si še en scenarij kjer $N = 4$, $D = 1$ in ceste povezujejo znamenitosti, kot je prikazano na spodnji sliki.



Ocenjevalnik pokliče funkcijo `longest_trip` na naslednji način:

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

V tem scenariju je najdaljši izlet dolg 2. Po nekaj klicih funkcije `are_connected` lahko funkcija `longest_trip` vrne enega izmed naslednjih odgovorov: $[0, 1]$, $[1, 0]$, $[2, 3]$ ali $[3, 2]$.

2. primer

Podnaloga 0 vsebuje dodaten testni scenarij z $N = 256$. Ta primer je vključen v priponki, ki jo lahko preneseš iz tekmovalnega sistema.

Omejitve

- $3 \leq N \leq 256$
- Vsota N med vsemi klici `longest_trip` ne presega $1\,024$ v nobenem testnem primeru.
- $1 \leq D \leq 3$

Podnaloge

1. (5 točk) $D = 3$
2. (10 točk) $D = 2$
3. (25 točk) $D = 1$. Naj $l^\star$ označuje dolžino najdaljšega izleta. Funkcija `longest_trip` ne rabi vrniti izleta dolžine $l^\star$. Namesto tega naj vrne izlet dolžine najmanj $\left\lceil \frac{l^\star}{2} \right\rceil$.
4. (60 točk) $D = 1$

Pri 4. podnalogi je končna ocena odvisna od števila klicev funkcije `are_connected` pri enem izvajanju funkcije `longest_trip`. Naj bo q največje število klicev med vsemi izvajanji `longest_trip` pri vsakem testnem primeru podnaloge. Končna ocena podnaloge se izračuna skladno s tabelo spodaj.

Pogoj	Točke
$2\,750 < q \leq 32\,640$	20
$550 < q \leq 2\,750$	30
$400 < q \leq 550$	45
$q \leq 400$	60

Če se pri kateremkoli testnem primeru klici funkcije `are_connected` ne skladajo z omejitvami opisanimi v razdelku Podrobnosti implementacije, ali pa je polje, ki ga vrne `longest_trip`, nepravilno, je končna ocena podnaloge 0.

Vzorčni ocenjevalnik

Naj C označuje število primerov - število klicev `longest_trip`. Vzorčni ocenjevalnik bere vhod oblike:

- 1. vrstica: C

Sledijo opisi C scenarijev.

Vzorčni ocenjevalnik bere opis vsakega scenarija naslednje oblike:

- vrstica 1: $N\ D$
- vrstice $1+i$, $(1 \leq i < N)$: $U_i[0]\ U_i[1]\ \ldots\ U_i[i-1]$

Vsak U_i ($1 \leq i < N$) je polje velikosti i, ki opisuje pare s cesto povezanih znamenitosti. Za vsak i in j, kjer $1 \leq i < N$ and $0 \leq j < i$:

- če sta znamenitosti i in j povezani s cesto, je vrednost $U_i[j] = 1$.
- če ni ceste med znamenitostima i in j, je vrednost $U_i[j] = 0$.

V vsakem primeru vzorčni ocenjevalnik še pred klicem `longest_trip` preveri, da je gostota cestnega omrežja vsaj D. Če to ne velja, izpiše `Insufficient Density` in zaključi izvajanje.

Če vzorčni ocenjevalnik zazna kršitev pravil, izpiše `Protocol Violation: <MSG>`, kjer je `<MSG>` eno izmed naslednjih sporočil:

- `invalid array`: pri klicu `are_connected` je vsaj eno izmed polj A ali B
  - prazno
  - vsebuje celico, ki ni celo število med 0 in vključno $N - 1$
  - vsebuje isto celico vsaj dvakrat
- `non-disjoint arrays`: pri klicu `are_connected` polji A in B nista disjunktni.
- `too many calls`: število klicev funkcije `are_connected` presega $32\,640$ za en klic funkcije `longest trip` ali skupno presega $150\,000$.
- `too many elements`: Skupno število znamenitosti pri klicu funkcije `are_connected` pri vseh klicih funkcije `longest trip` presega $1\,500\,000$.

Drugače naj bodo celice polja, ki ga vrne `longest_trip` za test, $t[0], t[1], \ldots, t[l-1]$ za nek nenegativen l. Vzorčni ocenjevalnik izpiše tri vrstice v naslednji obliki:

- vrstica 1: l
- vrstica 2: $t[0]\ t[1]\ \ldots\ t[l-1]$
- vrstica 3: število klicev funkcije `are_connected` v primeru

Na koncu vzorčni ocenjevalnik izpiše:

- vrstice $1 + 3 \cdot C$: Največje število klicev `are_connected` za vse klice `longest_trip`