



## Garākais ceļojums

IOI 2023 organizētājiem ir lielas nepatīkšanas! Viņi aizmirsuši saplānot ceļojumu uz Ópusztaszer nākamajai dienai. Bet varbūt vēl nav par vēlu...

Iekš Ópusztaszer ir  $N$  apskates vietas, kas sanumurētas ar skaitļiem no 0 līdz  $N - 1$ . Daži pāri ar šīm vietām ir savienotas ar *divvirziena* **ceļiem**. Katrs apskates vietu pāris ir savienots savā starpā ar ne vairāk kā vienu ceļu. Organizētāji *nezina*, kuras apskates vietas ir savienotas ar ceļiem.

Teiksim, ka Ópusztaszer ceļu tīkla **blīvums** ir **vismaz**  $\delta$ , ja katras 3 atšķirīgas apskates vietas satur vismaz  $\delta$  ceļus starp tām. Citiem vārdiem, katram apskates vietu trijniekam  $(u, v, w)$ , kur  $0 \leq u < v < w < N$ , starp vietu pāriem  $(u, v)$ ,  $(v, w)$  un  $(u, w)$  vismaz  $\delta$  pāri ir savienoti ar ceļu.

Organizētāji *zina* pozitīvu veselu skaitli  $D$  tādu, ka ceļu tīkla blīvums ir vismaz  $D$ . Ņemiet vērā, ka  $D$  vērtība nevar būt lielāka par 3.

Organizētāji var veikt **zvanus** Ópusztaszeri tālruna dispečeram, lai iegūtu informāciju par ceļiem starp noteiktām apskates vietām. Katrā zvanā jāsniedz divi netukši apskates vietu masīvi  $[A[0], \dots, A[P - 1]]$  un  $[B[0], \dots, B[R - 1]]$ . Visām šīm vietām jābūt dažādām, tas ir,

- $A[i] \neq A[j]$  visiem  $i$  un  $j$ , kuriem  $0 \leq i < j < P$ ;
- $B[i] \neq B[j]$  visiem  $i$  un  $j$ , kuriem  $0 \leq i < j < R$ ;
- $A[i] \neq B[j]$  visiem  $i$  and  $j$ , kuriem  $0 \leq i < P$  un  $0 \leq j < R$ .

Katram zvanam dispečers ziņo, vai eksistē ceļš starp kādu vietu no masīva  $A$  un kādu vietu no masīva  $B$ . Precīzāk, dispečers pārbauda visus pārus  $i$  un  $j$ , kuriem  $0 \leq i < P$  un  $0 \leq j < R$ . Ja kādam no tiem vietas  $A[i]$  un  $B[j]$  ir savienotas ar ceļu, dispečers atgriež `true`. Citādi, dispečers atgriež `false`.

**Ceļojums** garumā  $l$  ir virkne ar *dažādām* apskates vietām  $t[0], t[1], \dots, t[l - 1]$ , kur katram  $i$  no 0 līdz  $l - 2$ , ieskaitot, vieta  $t[i]$  un vieta  $t[i + 1]$  ir savienotas ar ceļu. Ceļojums garumā  $l$  tiek saukts par **garāko ceļojumu**, ja neeksistē ceļojums garumā vismaz  $l + 1$ .

Jūsu uzdevums ir palīdzēt organizētājiem atrast kādu garāko Ópusztaszer ceļojumu, veicot zvanus dispečerim.

## Implementācijas detaļas

Jums ir jāimplementē šāda procedūra:

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

- $N$ : Ópusztaszer apskates vietu skaits.
- $D$ : garantēts minimālais ceļu tīkla blīvums.
- Šai procedūrai jāatgriež masīvs  $t = [t[0], t[1], \dots, t[l - 1]]$ , kas apraksta garāko ceļojumu.
- Šī procedūra var tikt izsaukta **vairākas reizes** katram testam.

Augšā aprakstītā procedūra var izsaukt šādu procedūru:

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

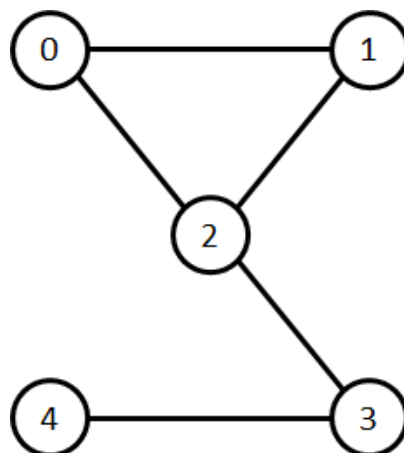
- $A$ : netukšs masīvs ar dažādām apskates vietām.
- $B$ : netukšs masīvs ar dažādām apskates vietām.
- $A$  un  $B$  jāsastāv no dažādiem skaitļiem.
- Šī procedūra atgriež `true`, ja eksistē vieta no  $A$  un vieta no  $B$ , kas ir savienotas ar ceļu. Citādi, tā atgriež `false`.
- Šī procedūra var tikt izsaukta ne vairāk kā 32 640 reizes katrā `longest_trip` izsaukumā, un ne vairāk kā 150 000 reizes kopā.
- Kopējais masīvu  $A$  un  $B$  garums, kas sniegti šai procedūrai pāri visiem tās izsaukumiem nevar pārsniegt 1 500 000.

Vertētājs ir **neadaptīvs**. Katrs iesūtījums tiek vērtēts uz vienas un tās pašas testu kopas. Tas nozīmē, ka vērtības  $N$  un  $D$ , kā arī apskates vietu pāri, kas savienoti ar ceļiem, ir fiksēti katram `longest_trip` izsaukumam iekš katra testa.

## Piemēri

### 1. piemērs

Aplūkosim scenāriju, kur  $N = 5$ ,  $D = 1$ , un ceļu savienojumi ir parādīti šajā zīmējumā:



Procedūra `longest_trip` tiek izsaukta šādā veidā:

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

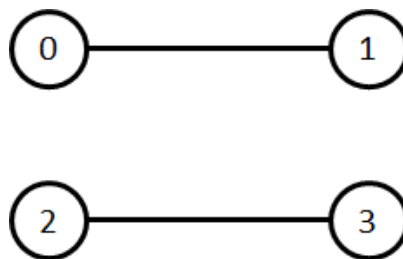
Procedūra var izsaukt procedūru `are_connected` šādi:

Izsaukums	Pāri, kas savienoti ar ceļu	Atgrieztā vērtība
<code>are_connected([0], [1, 2, 4, 3])</code>	(0,1) un (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>	nav	false

Pēc ceturtā izsaukuma, izrādās, ka *neviens* no pāriem (1,4), (0,4), (1,3) un (0,3) ir savienots ar ceļu. Tā kā ceļa tīkla blīvums ir vismaz  $D = 1$ , kļūst skaidrs, ka no trijnieka (0,3,4), pārim (3,4) jābūt savienotam ar ceļu. Līdzīgi, vietām 0 un 1 jābūt savienotām.

Šajā brīdī var secināt, ka  $t = [1, 0, 2, 3, 4]$  ir ceļojums garumā 5, un neeksistē ceļojums, kas ir garāks par 5. Līdz ar to, procedūra `longest_trip` var atgriezt `[1, 0, 2, 3, 4]`.

Aplūkosim citu scenāriju, kurā  $N = 4$ ,  $D = 1$ , un ceļi starp apskates vietām ir parādīti šajā zīmējumā:



Procedūra `longest_trip` tiek izsaukta šādi:

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

Šajā scenārijā, garākā ceļojumā garums ir 2. Līdz ar to, pēc dažiem procedūras `are_connected` izsaukumiem, procedūra `longest_trip` var atgriezt vienu no `[0, 1]`, `[1, 0]`, `[2, 3]` vai `[3, 2]`.

## Piemērs 2

0. apakšuzdevums satur papildus testu ar  $N = 256$  apskates vietām. Šis tests ir iekļauts failu arhīvā, kuru jūs varat lejupielādēt no sacensību sistēmas.

## Ierobežojumi

- $3 \leq N \leq 256$
- Visu  $N$  summa pāri visiem `longest_trip` izsaukumiem nepārsniedz 1 024 katrā testā.
- $1 \leq D \leq 3$

## Apakšuzdevumi

1. (5 punkti)  $D = 3$
2. (10 punkti)  $D = 2$
3. (25 punkti)  $D = 1$ . Ar  $l^*$  apzīmēsim garākā ceļojuma garumu. Procedūrai `longest_trip` nav obligāti jāatgriež ceļojumu garumā  $l^*$ . Tā vietā, tai ir jāatgriež ceļojums ar garumu vismaz  $\left\lceil \frac{l^*}{2} \right\rceil$ .
4. (60 punkti)  $D = 1$

4. apakšuzdevumā jūsu vērtējums tiek aprēķināts, balstoties uz procedūras `are_connected` izsaukumu skaitu pāri vienam `longest_trip` izsaukumam. Ar  $q$  apzīmēsim lielāko `are_connected` izsaukumu skaitu starp visiem `longest_trip` izsaukumiem pāri visiem apakšuzdevuma testiem. Jūsu vērtējums šim apakšuzdevumam tiek aprēķināts atbilstoši šai tabulai:

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

Ja kādā testā izsaukumi procedūrai `are_connected` neatbilst ierobežojumiem, kas aprakstīti sadaļā Implementācijas detaļas, vai arī masīvs, ko atgriež `longest_trip` ir nepareizs, jūsu risinājuma vērtējums šim apakšuzdevumam būs vienāds ar 0.

## Piemēra vērtētājprogramma

Ar  $C$  apzīmēsim scenāriju skaitu, tas ir, `longest_trip` izsaukumu skaitu. Piemēra vērtētājprogramma nolasa ievadu šādā formātā:

- 1. rinda:  $C$

Tad seko  $C$  scenāriju apraksti.

Piemēra vērtētājprogramma nolasa katra scenārija aprakstu šādā formātā:

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

Šeit, katrs  $U_i$  ( $1 \leq i < N$ ) ir masīvs ar izmēru  $i$ , kas apraksta, kuri pāri ar apskates vietām ir savienoti ar ceļiem. Visiem  $i$  un  $j$ , kuriem  $1 \leq i < N$  un  $0 \leq j < i$ :

- ja vietas  $j$  un  $i$  ir savienotas ar ceļu, tad  $U_i[j]$  jābūt 1;
- ja nav ceļa, kas savieno vietas  $j$  un  $i$ , tad  $U_i[j]$  jābūt 0.

Katrā scenārijā, pirms `longest_trip` izsaukšanas, piemēra vērtētājprogramma pārbauda, vai ceļu tīkla blīvums ir vismaz  $D$ . Ja šis nosacījums neizpildās, tā izdrukā ziņojumu `Insufficient Density` un izbeidz darbību.

Ja piemēra vērtētājprogramma nosaka protokola pārkāpumu, tās izvads ir `Protocol Violation: <MSG>`, kur `<MSG>` ir viens no šiem ziņojumiem:

- `invalid array: are_connected` izsaukumā vismaz viens no masīviem  $A$  un  $B$ 
  - ir tukšs vai
  - satur elementu, kas nav vesels skaitlis starp 0 un  $N - 1$ , ieskaitot, vai
  - satur kādu elementu vismaz divreiz.
- `non-disjoint arrays: are_connected` izsaukumā, masīvi  $A$  un  $B$  nesastāv no dažādiem skaitļiem.
- `too many calls`: izsaukumu skaits procedūrai `are_connected` pārsniedz 32 640 pāri pašreizējam `longest_trip` izsaukumam vai pārsniedz 150 000 kopumā.
- `too many elements`: kopējais apskates vietu skaits, kas sniegtas procedūrai `are_connected` pāri visiem izsaukumiem pārsniedz 1 500 000.

Citādi, apzīmēsim `longest_trip` atgrieztā masīva elementus kādā scenārijā ar  $t[0], t[1], \dots, t[l - 1]$ , kādam nenegatīvam  $l$ . Piemēra vērtētājprogramma izdrukā trīs rindiņas šim scenārijam šādā formātā:

- 1. rinda:  $l$
- 2. rinda:  $t[0] \ t[1] \ \dots \ t[l - 1]$
- 3. rinda: `are_connected` izsaukumu skaits šajā scenārijā

Beigās piemēra vērtētājprogramma izdrukā:

- $(1 + 3 \cdot C)$ -tā rinda: lielākais `are_connected` izsaukumu skaits pāri visiem `longest_trip` izsaukumiem.