



## Pikim reis

IOI 2023 korraldajad on suures jamas! Nad unustasid planeerida ekskursiooni Ópusztaszerisse. Äkki pole veel liiga hilja...

Ópusztaszeris on  $N$  vaatamisväärsust, nummerdatud 0 kuni  $N - 1$ . Mõned vaatamisväärsuste paarid on ühendatud *kahesuunaliste teedega*. Iga vaatamisväärsuste paari vahel on ülimalt üks tee. Korraldajad *ei tea*, missuguste vaatamisväärsuste vahel on teed.

Ütleme, et Ópusztaszeri teedevõrgu **tihedus** on **vähemalt**  $\delta$ , kui iga 3 eri vaatamisväärsuse vahel on vähemalt  $\delta$  teed. Teiste sõnadega on iga vaatamisväärsuste kolmiku  $(u, v, w)$  jaoks (kus  $0 \leq u < v < w < N$ ) paaride  $(u, v)$ ,  $(v, w)$  ja  $(u, w)$  hulgas vähemalt  $\delta$  paari vahel teed.

Korraldajad *teavad* positiivset täisarvu  $D$ , mis näitab, et teedevõrgu tihedus on vähemalt  $D$ . Pane tähele, et  $D$  väärtus ei saa olla suurem kui 3.

Korraldajad saavad **helistada** Ópusztaszeri telefonioperaatorile, et saada infot kindlate vaatamisväärsuste vaheliste teede kohta. Igas kõnes peab välja tooma kaks mittetühja vaatamisväärsuste massiivi  $[A[0], \dots, A[P - 1]]$  ja  $[B[0], \dots, B[R - 1]]$ . Vaatamisväärsused peavad olema paarikaupa erinevad, s.t.

- $A[i] \neq A[j]$  iga  $i$  ja  $j$  korral, kus  $0 \leq i < j < P$ ;
- $B[i] \neq B[j]$  iga  $i$  ja  $j$  korral, kus  $0 \leq i < j < R$ ;
- $A[i] \neq B[j]$  iga  $i$  ja  $j$  korral, kus  $0 \leq i < P$  ja  $0 \leq j < R$ .

Igas kõnes annab operaator teada, kas leidub tee, mis ühendab mingit vaatamisväärsust hulgast  $A$  mingi vaatamisväärsusega hulgast  $B$ . Täpsemini käib operaator läbi kõik  $0 \leq i < P$  ja  $0 \leq j < R$  paarid ning kui leidub paar, kus  $A[i]$  ja  $B[j]$  vahel on tee, siis tagastab operaator `true` ja vastasel juhul `false`.

**Reis** pikkusega  $l$  on *paarikaupa erinevate* vaatamisväärsuste jada  $t[0], t[1], \dots, t[l - 1]$ , kus iga  $i$  jaoks vahemikus 0 ja  $l - 2$  (kaasaarvatud) on vaatamisväärsuste  $t[i]$  ja  $t[i + 1]$  vahel tee. Reisi pikkusega  $l$  nimetame **pikimaks reisiks**, kui ei leidu reisi pikkusega  $l + 1$ .

Sinu ülesanne on operaatorile kõnesid tehes aidata korraldajatel leida pikim reis Ópusztaszeris.

## Realisatsioon

Lahendusena tuleb realiseerida funktsioon

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

- $N$ : vaatamisväärsuste arv Ópusztaszeris.
- $D$ : teedevõrgu garanteeritud miinimumtihedus.
- Funktsioon peab tagastama pikimat reisi väljendava massiivi  $t = [t[0], t[1], \dots, t[l - 1]]$ .
- Seda funktsiooni võidakse ühes testis käivitada **korduvalt**.

Ülaltoodud funktsioon võib kutsuda välja funktsiooni

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

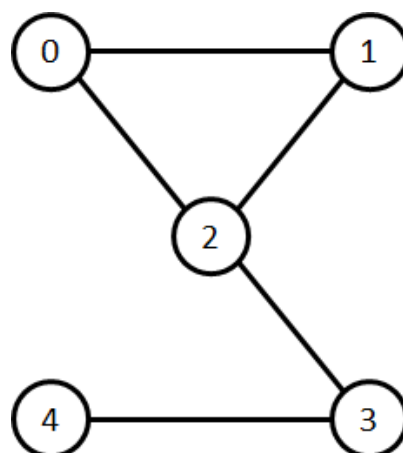
- $A$ : mittetühi erinevate vaatamisväärsuste massiiv.
- $B$ : mittetühi erinevate vaatamisväärsuste massiiv.
- $A$  ja  $B$  peavad olema ühisosata.
- Funktsioon tagastab `true`, kui leiduvad kaks vaatamisväärsust, üks hulgast  $A$  ja teine hulgast  $B$ , mille vahel on tee. Vastasel juhul tagastab see `false`.
- Funktsiooni `are_connected` võib välja kutsuda ülimalt 32 640 korda iga funktsiooni `longest_trip` väljakutse kohta ja kokku ülimalt 150 000 korda.
- Sellele funktsioonile üle kõigi väljakutsete antud massiivide  $A$  ja  $B$  kogupikkus ei tohi ületada 1 500 000.

Hindaja **ei ole adaptiivne**. Iga esitust hinnatakse samade testidega. Kõigis testides on  $N$  ja  $D$  väärtused ning omavahel teedega ühendatud vaatamisväärsuste paarid on kindlaks määratud enne funktsiooni `longest_trip` iga käivitust.

## Näited

### Näide 1

Vaatame juhtu, kus  $N = 5$ ,  $D = 1$  ja teedevõrk on toodud järgmisel joonisel:



Funktsioon `longest_trip` kutsutakse välja järgmisel viisil:

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

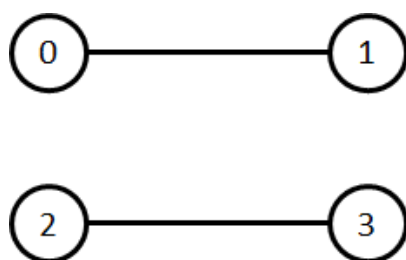
Funktsioon `longest_trip` võib kutsuda funktsiooni `are_connected` järgmisel viisil:

Väljakutse	Paarid, mille vahel on teed	Tagastatav väärtus
<code>are_connected([0], [1, 2, 4, 3])</code>	(0,1) ja (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>	puuduvad	false

Pärast neljandat väljakutset on selge, et *mitte ükski* paaridest (1,4), (0,4), (1,3) ja (0,3) ei ole teega ühendatud. Kuna teedevõrgu tihedus on vähemalt  $D = 1$ , siis näeme, et kolmikust (0,3,4) peab paari (3,4) vahel olema tee. Sarnaselt peab vaatamisväärsuste 0 ja 1 vahel olema tee.

Nüüd saab järeldada, et  $t = [1, 0, 2, 3, 4]$  on reis pikkusega 5, ja pikemat reisi ei ole olemas. Seega võib `longest_trip` tagastada `[1, 0, 2, 3, 4]`.

Vaatame teist juhtu, kus  $N = 4$ ,  $D = 1$  ja vaatamisväärsuste vahelised teed on toodud järgmisel joonisel:



Funktsiooni `longest_trip` kutsutakse välja järgmiselt:

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

Selles olukorras on pikima reisi pikkus 2. Seega võib pärast mõningaid funktsiooni `are_connected` väljakutseid funktsioon `longest_trip` tagastada kas `[0, 1]`, `[1, 0]`, `[2, 3]` või `[3, 2]`.

## Näide 2

Alamülesandes 0 on ka teine näidistest, kus on  $N = 256$  vaatamisväärsust. See test on leitav ülesande abifailide pakis, mille saab alla laadida võistlussüsteemist.

## Piirangud

- $3 \leq N \leq 256$ .
- Väärtuste  $N$  summa üle kõigi `longest_trip` väljakutsete ei ületa üheski testis 1 024.
- $1 \leq D \leq 3$ .

## Alamülesanded

1. (5 punkti)  $D = 3$ .
2. (10 punkti)  $D = 2$ .
3. (25 punkti)  $D = 1$ . Olgu  $l^*$  pikima reisi pikkus. Funktsioon `longest_trip` ei pea tagastama reisi pikkusega  $l^*$ . Piisab, kui tagastada reis pikkusega vähemalt  $\left\lceil \frac{l^*}{2} \right\rceil$ .
4. (60 punkti)  $D = 1$ .

Alamülesandes 4 määrab punktisumma funktsiooni `are_connected` väljakutsete arv ühe `longest_trip` väljakutse jooksul. Olgu  $q$  maksimaalne `longest_trip` väljakutsete arv iga testjuhu jaoks selles alamülesandes. Selle alamülesande eest saadud punktisumma arvutatakse järgmiselt:

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

Kui mõne testjuhu korral ei vasta `are_connected` väljakutsete arv osas "Realisatsioon" kirjeldatud piirangutele või kui `longest_trip` poolt tagastatud massiiv on väär, saab lahendus vastava alamülesande eest 0 punkti.

## Hindamisprogramm

Olgu  $C$  juhtumite arv, s.t. funktsiooni `longest_trip` väljakutsete arv. Hindamisprogramm loeb sisendit järgmises vormingus:

- rida 1:  $C$

Järgnevad  $C$  juhtumi kirjeldused.

Hindamisprogramm loeb iga juhtumi kirjeldust järgmises vormingus:

- rida 1:  $N$   $D$
- rida  $1 + i$  (kus  $1 \leq i < N$ ):  $U_i[0]$   $U_i[1]$  ...  $U_i[i - 1]$

Iga  $U_i$  ( $1 \leq i < N$ ) on massiiv pikkusega  $i$ , mis kirjeldab teedega ühendatud vaatamisväärsuste paare. Iga  $i$  ja  $j$  jaoks, kus  $1 \leq i < N$  ja  $0 \leq j < i$ :

- kui vaatamisväärsuste  $j$  ja  $i$  vahel on tee, siis on  $U_i[j]$  väärtus 1;
- kui vaatamisväärsuste  $j$  ja  $i$  vahel ei ole teed, siis on  $U_i[j]$  väärtus 0.

Iga juhtumi korral kontrollib hindamisprogramm enne `longest_trip` väljakutset, kas teedevõrgu tihedus on vähemalt  $D$ . Kui ei, siis kuvatakse sõnum `Insufficient Density` ja programm lõpetab töö.

Kui hindamisprogramm tuvastab suhtlusprotokolli rikkumise, siis on hindamisprogrammi väljund `Protocol Violation: <MSG>`, kus `<MSG>` on üks järgmistest veateadetest:

- `invalid array`: funktsiooni `are_connected` väljakutses on vähemalt üks massiividest  $A$  ja  $B$  kas
  - tühi,
  - sisaldab elementi, mis ei ole täisarv lõigust 0 kuni  $N - 1$  (kaasa arvatud) või
  - sisaldab sama elementi vähemalt kaks korda.
- `non-disjoint arrays`: funktsiooni `are_connected` väljakutses ei ole  $A$  ja  $B$  ühisosata.
- `too many calls`: funktsiooni `are_connected` väljakutsete arv ületab ühe `longest_trip` väljakutse jaoks 32 640 või kokkuvõttes 150 000.
- `too many elements`: funktsioonile `are_connected` sisendiks antud vaatamisväärsuste koguarv on suurem kui 1 500 000.

Vastasel juhul olgu funktsiooni `longest_trip` tagastatava massiivi elemendid selle juhtumi jaoks  $t[0], t[1], \dots, t[l - 1]$ , kus  $l$  on mittenegatiivne. Hindamisprogramm kuvab selle juhtumi jaoks kolm rida järgmises vormingus:

- rida 1:  $l$
- rida 2:  $t[0] \ t[1] \ \dots \ t[l - 1]$
- rida 3: funktsiooni `are_connected` väljakutsete arv selle juhtumi jaoks.

Lõpuks kuvab näidishindaja:

- rida  $1 + 3 \cdot C$ : suurim `are_connected` väljakutsete arv üle kõigi `longest_trip` väljakutsete.