



Ilgiausias maršrutas

IOI 2023 organizatoriai turi rūpesčių! Jie užmiršo suorganizuoti rytojaus išvyką į Ópusztaszer etnografinį kaimą. Bet gal dar ne per vėlu ...

Ópusztaszer kaime yra N lankytinų objektų, sunumeruotų nuo 0 iki $N - 1$. Kai kurias objektų poras jungia *dvikrypčiai keliai*. Bet kurių objektų porą jungia ne daugiau nei vienas kelias. IOI organizatoriai *nežino*, kurie objektai sujungti keliais.

Sakysime, kad Ópusztaszer kelių tinklo **tankumas** yra **mažiausiai** δ , jei bet kurie 3 skirtingi objektai turi ne mažiau kaip δ juos jungiančių kelių. Kitaip sakant, bet kuriam lankytinų objektų trejetui (u, v, w) , kuriam galioja $0 \leq u < v < w < N$, bent δ porų yra sujungtos keliais (galimos poros: (u, v) , (v, w) ir (u, w)).

Organizatoriai *žino*, kad D , kad kelių tinklo tankumas yra mažiausiai D (tai teigiamas sveikasis skaičius). Atkreipkite dėmesį, kad D reikšmė negali būti didesnė už 3.

Organizatoriai gali **skambinti** į Ópusztaszer dispečerinę ir gauti informaciją apie kelius tarp tam tikrų objektų. Kiekvieno skambučio metu nurodomi du netušti lankytinų objektų masyvai: $[A[0], \dots, A[P - 1]]$ ir $[B[0], \dots, B[R - 1]]$. Masyvuose nurodyti lankytini objektai turi būti poromis skirtingi:

- $A[i] \neq A[j]$ visiems i ir j , kuriems galioja $0 \leq i < j < P$;
- $B[i] \neq B[j]$ visiems i ir j , kuriems galioja $0 \leq i < j < R$;
- $A[i] \neq B[j]$ visiems i ir j , kuriems galioja $0 \leq i < P$ ir $0 \leq j < R$.

Kiekvieno skambučio metu dispečeris pateikia informaciją, ar yra kelias, jungiantis nors vieną objektą iš masyvo A su nors vienu objektu iš masyvo B . Tai reiškia, kad dispečeris patikrina visas galimas poras i ir j kur $0 \leq i < P$, $0 \leq j < R$, ir grąžina `true`, jei egzistuoja tokie i ir j , kuriems objektus $A[i]$ ir $B[j]$ jungia kelias. Priešingu atveju dispečeris grąžina `false`.

Maršrutas, kurio ilgis l , yra *skirtingų* lankytinų objektų seka $t[0], t[1], \dots, t[l - 1]$, kur kiekvienam i nuo 0 iki $l - 2$ imtinai, objektai $t[i]$ ir $t[i + 1]$ yra sujungti keliais. l ilgio maršrutas vadinamas **ilgiausiu maršrutu**, jei neegzistuoja maršrutas, kurio ilgis yra bent $l + 1$.

Skambindami ir pateikdami užklausas dispečeriui, padėkite organizatoriams rasti ilgiausią maršrutą po Ópusztaszer kaimą.

Realizacija

Parašykite šią funkciją:

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

- N : lankytinų objektų skaičius Ópusztaszer etnografiniame kaime.
- D : garantuotas mažiausias kelių tinklo tankis.
- Ši funkcija turi grąžinti masyvą, nusakantį ilgiausią maršrutą $t = [t[0], t[1], \dots, t[l - 1]]$.
- Ši funkcija gali būti iškviesta **kelis kartus** kiekvienam testui.

Ši funkcija gali kviesti žemiau aprašytą funkciją:

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

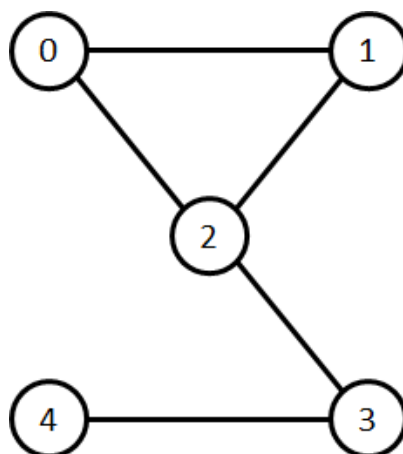
- A : netuščias skirtingų lankytinų objektų masyvas.
- B : netuščias skirtingų lankytinų objektų masyvas.
- A ir B turi neturėti bendrų elementų.
- Ši funkcija grąžina `true`, jei yra objektas masyve A ir objektas masyve B , kuriuos jungia kelias. Kitu atveju grąžina `false`.
- Ši funkcija gali būti iškviesta ne daugiau 32 640 kartų kiekviename funkcijos `longest_trip` iškvietime, ir ne daugiau 150 000 kartų iš viso.
- Bendras masyvų A ir B , pateiktų funkcijai per visus jos iškvietimus, ilgis negali viršyti 1 500 000.

Vertinimo programa **nėra prisitaikanti**. Kiekviena pateikta programa testuojama su tuo pačiu testų rinkiniu. Tai yra, N ir D reikšmės, o taip pat objektų poros, kurias jungia keliai, yra fiksuotos kiekvienam `longest_trip` iškvietimui kiekviename teste.

Pavyzdžiai

Pavyzdys nr. 1

Tarkime, kad $N = 5$, $D = 1$, o kelių tinklas parodytas paveikslėlyje žemiau:



Funkcija `longest_trip` kviečiama taip:

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

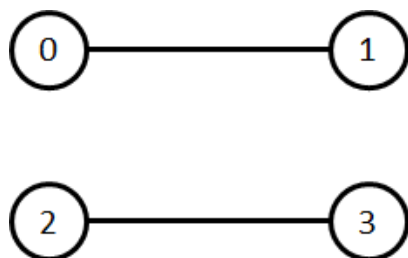
Ši funkcija gali atlikti tokius `are_connected` iškvietimus.

Iškvietimas	Poros, kurias jungia kelias	Grąžinama reikšmė
<code>are_connected([0], [1, 2, 4, 3])</code>	(0,1) ir (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>	nėra	false

Po ketvirtojo iškvietimo paaiškėja, kad *nei viena iš porų* (1,4), (0,4), (1,3) ir (0,3) nėra sujungta keliu. Kadangi kelių tinklo tankumas yra ne mažesnis nei $D = 1$, matome, kad trejeto (0,3,4) porą (3,4) būtinai jungia kelias. Analogiškai gauname, kad kelias jungia objektus 0 ir 1.

Jau galima padaryti išvadą, kad $t = [1, 0, 2, 3, 4]$ yra maršrutas ir jo ilgis yra 5, ir nėra maršruto, kuris būtų ilgesnis nei 5. Taigi, funkcija `longest_trip` gali grąžinti `[1, 0, 2, 3, 4]`.

Įsivaizduokime kitą scenarijų, kuriame $N = 4$, $D = 1$, o objektus jungiantys keliai pavaizduoti paveikslėlyje žemiau:



Funkcija `longest_trip` iškviečiama taip:

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

Šiuo atveju ilgiausio maršruto ilgis lygus 2. Tad po kelių funkcijos `are_connected` iškvietimų funkcija `longest_trip` gali gražinti vieną iš šių atsakymų: `[0, 1]`, `[1, 0]`, `[2, 3]` arba `[3, 2]`.

Pavyzdys nr. 2

Dalinėje užduotyje nr. 0 įdėtas papildomas pavyzdinis testas su $N = 256$ lankytiniais objektais. Šis testas pateiktas priede, kurį galima parsisiųsti iš varžybų sistemos.

Ribojimai

- $3 \leq N \leq 256$
- Kiekvienam testui skaičių N suma per visus `longest_trip` iškvietus viename teste neviršija 1 024.
- $1 \leq D \leq 3$

Daliniai ribojimai

1. (5 taškai) $D = 3$
2. (10 taškų) $D = 2$
3. (25 taškai) $D = 1$. Ilgiausio maršruto ilgį pažymėkime l^* . Funkcija `longest_trip` neturi grąžinti l^* ilgio maršruto. Ji turi grąžinti maršrutą, kurio ilgis bent $\left\lceil \frac{l^*}{2} \right\rceil$.
4. (60 taškų) $D = 1$

Taškų skaičius už dalinę užduotį nr. 4 apskaičiuojamas pagal funkcijos `are_connected` iškviety skaičių, gaunamų iš vieno `longest_trip` iškviety. Pažymėkime q didžiausią iškviety skaičių tarp visų `longest_trip` iškviety tarp visų dalinės užduoties testų.

Taškai už šią dalinę užduotį apskaičiuojami remiantis žemiau pateikta lentele:

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

Jei nors viename teste funkcijos `are_connected` iškviety neatitinka Realizacijos skyrelyje pateiktų reikalavimų, arba jei `longest_trip` grąžina neteisingą masyvą, už šią dalinę užduotį skiriama 0 taškų.

Pavyzdinė vertinimo programa

Pažymėkime C scenarijų skaičių (`longest_trip` iškviety kiekį). Pavyzdinė vertinimo programa duomenis skaito šiuo formatu:

- 1-a eilutė: C

Toliau pateikta C scenarijų.

Pavyzdinė vertinimo programa perskaito kiekvieną scenarijų tokiu formatu:

- 1-a eilutė: $N \ D$
- $(1 + i)$ -a eilutė ($1 \leq i < N$): $U_i[0] \ U_i[1] \ \dots \ U_i[i - 1]$

Čia kiekvienas U_i ($1 \leq i < N$) yra i dydžio masyvas, nusakantis, kurios lankytinų objektų poros yra sujungtos keliu. Kiekvienam i ir j , kur $1 \leq i < N$ ir $0 \leq j < i$:

- jei lankytini objektai j ir i yra sujungti keliu, tai $U_i[j]$ vertė turėtų būti 1;
- jei nėra kelio, jungiančio lankytinus objektus j ir i , tada $U_i[j]$ vertė turėtų būti 0.

Kiekviename scenarijuje prieš iškviečiant `longest_trip` pavyzdinė vertinimo programa patikrina, ar kelių tinklo tankumas yra bent D . Jei ši sąlyga negalioja, išveda žinutę `Insufficient Density` ir programa baigia darbą.

Jei pavyzdinė vertinimo programa aptinka protokolo pažeidimą, išveda žinutę `Protocol Violation: <MSG>`, kur `<MSG>` yra viena iš šių klaidos žinučių:

- `invalid array: are_connected` iškvietime bent vienas iš masyvų A ir B
 - yra tuščias, arba
 - turi narį, kuris nėra sveikasis skaičius nuo 0 iki $N - 1$ imtinai, arba
 - tas pats narys masyve pasikartoja bent du kartus.
- `non-disjoint arrays: are_connected` iškvietime masyvai A ir B turi bent vieną bendrą narį.
- `too many calls: are_connected` iškvietimų kiekis viršija 32 640 šiame `longest_trip` funkcijos iškvietime, arba viršija 150 000 per visus `longest_trip` iškvietimus.
- `too many elements: are_connected` funkcijai pateiktų lankytinų objektų kiekis viršija 1 500 000 per visus iškvietimus.

Kitu atveju, pažymėkime `longest_trip` scenarijuje grąžintus narius $t[0], t[1], \dots, t[l - 1]$ kažkokiam neneigiamam l . Pavyzdinė vertinimo programa išveda tris eilutes šiuo formatu:

- 1-a eilutė: l
- 2-a eilutė: $t[0] \ t[1] \ \dots \ t[l - 1]$
- 3-a eilutė: funkcijos `are_connected` iškvietimų skaičius šiame scenarijuje.

Galiausiai pavyzdinė vertinimo programa išveda:

- $(1 + 3 \cdot C)$ -a eilutė: didžiausią funkcijos `are_connected` iškvietimų skaičių tarp visų `longest_trip` iškvietimų.