



Leghosszabb út (Longest Trip)

Az IOI 2023 szervezői nagy bajban vannak! Elfelejtették megtervezni az Ópusztaszerre tervezett kirándulást a következő napra. De talán még nem késő ...

Ópusztaszeren N helyszín található, 0-tól $N - 1$ -ig számozva. Ezen helyszínek közt néhány párt *kétirányú út* köt össze. Bármely két helyszínt legfeljebb egy út köt össze. A szervezők *nem tudják*, hogy mely helyszíneket kötnék össze utak.

Azt mondjuk, hogy Ópusztaszer úthálózatának **sűrűsége legalább δ** , ha bármely 3 különböző helyszín közt legalább δ út van. Más szavakkal, minden olyan (u, v, w) hármas esetén, ahol $0 \leq u < v < w < N$, az (u, v) , a (v, w) és az (u, w) helyszínpárok közül legalább δ pár közt van út.

A szervezők *tudnak* egy olyan D pozitív egész számot, hogy az úthálózat sűrűsége legalább D . Megjegyzés: a D értéke nem lehet nagyobb, mint 3.

A szervezők **felhívhatják** Ópusztaszer telefonos diszpécserét, hogy információt szerezzenek az egyes helyszínek közti utakról. Minden egyes hívásban meg kell adni két, nem üres, helyszíneket tartalmazó $[A[0], \dots, A[P - 1]]$ és $[B[0], \dots, B[R - 1]]$ tömböt. A helyszíneknek páronként különbözőeknek kell lenniük, azaz,

- $A[i] \neq A[j]$ minden olyan i és j esetén, ahol $0 \leq i < j < P$;
- $B[i] \neq B[j]$ minden olyan i és j esetén, ahol $0 \leq i < j < R$;
- $A[i] \neq B[j]$ minden olyan i és j esetén, ahol $0 \leq i < P$ és $0 \leq j < R$.

A diszpécser minden egyes hívásnál megmondja, hogy van-e olyan út, amely összeköti valamely A -beli és valamely B -beli helyszíneket. Vagyis a diszpécser "true" értéket ad vissza, ha létezik olyan i és j , ahol $0 \leq i < P$ és $0 \leq j < R$, és $A[i]$ és $B[j]$ között van út. Ellenkező esetben a diszpécser "false" értéket ad vissza.

Egy l hosszúságú **útvonal** a $t[0], t[1], \dots, t[l - 1]$ *különböző* helyszínek sorozata, ahol a 0 és az $l - 2$ közötti minden i esetében (beleértve a határokat is) a $t[i]$ és a $t[i + 1]$ helyszínt út köti össze. Egy l hosszúságú útvonalat **leghosszabb útvonalnak** nevezünk, ha nem létezik legalább $l + 1$ hosszúságú útvonal.

A feladatod, hogy a diszpécserrel való telefonálással segíts a szervezőknek megtalálni a leghosszabb útvonalat Ópusztaszeren.

Megvalósítás

A következő eljárást kell megvalósítanod:

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

- N : az Ópusztaszeren található helyszínek száma.
- D : az úthálózat garantált minimális sűrűsége.
- Ennek a függvénynek egy $t = [t[0], t[1], \dots, t[l-1]]$ tömböt kell visszaadnia, amely egy leghosszabb útvonalat ír le.
- Ezt a függvényt minden egyes tesztesetben **többször** is hívhatják.

A fenti függvény meghívhatja a következő függvényt:

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

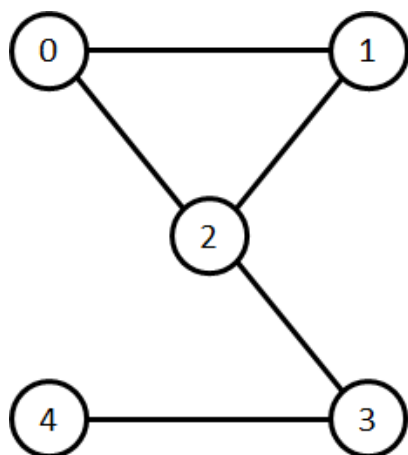
- A : a különböző helyszínek nem üres tömbje.
- B : különböző helyszínek nem üres tömbje.
- A és B diszjunkt halmazok.
- Ez az eljárás "true" értéket ad vissza, ha létezik egy helyszín az A -ban és egy helyszín a B -ben, amit út köt össze. Ellenkező esetben false-t ad vissza.
- Ezt az eljárást a `longest_trip` minden egyes meghívásakor legfeljebb 32 640-szer, összesen pedig legfeljebb 150 000-szer hívhatja az értékelő.
- A függvénynek átadott A és B tömbök teljes hossza - az összes hívás során - nem haladhatja meg az 1 500 000 értéket.

Az értékelő **nem adaptív**. Minden beadást ugyanazon tesztesetek alapján értékeli. Ez azt jelenti, hogy az N és D értékek, valamint az utakkal összekapcsolt helyszínek az egyes tesztesetekeken belül a `longest_trip` minden egyes hívásakor előre meghatározottak.

Példák

Példa 1

Nézzünk egy $N = 5$, $D = 1$ esetet, ahol a helyszínek közötti utak a következő ábrán láthatóak:



A `longest_trip` függvény hívása:

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

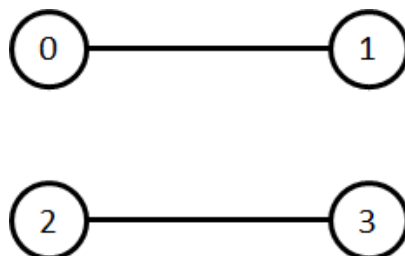
Az `are_connected` függvényhívások a következők:

Hívás	Úttal összekötött pontpárok	Visszatérési érték
<code>are_connected([0], [1, 2, 4, 3])</code>	(0,1) és (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>	nincs	false

A negyedik hívás után kiderül, hogy az (1,4), (0,4), (1,3) és (0,3) párok közül *egy*et sem köt össze út. Mivel a hálózat sűrűsége legalább $D = 1$, láthatjuk, hogy a (0,3,4) hármas és a (3,4) pár közt útnak kell vezetnie. Ehhez hasonlóan a 0 és 1 helyszínek között is van út.

Ezen a ponton megállapítható, hogy $t = [1, 0, 2, 3, 4]$ egy 5 hosszúságú út, és hogy nem létezik 5-nél hosszabb út. Ezért a `longest_trip` függvény a $[1, 0, 2, 3, 4]$ -t adja vissza.

Nézzünk egy $N = 4$, $D = 1$ esetet, ahol a helyszínek közötti utak a következő ábrán láthatóak:



A `longest_trip` függvény hívása:

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

Ebben az esetben a leghosszabb út hossza 2. Az `are_connected` eljárás néhány hívása után a `longest_trip` visszatérhet a $[0, 1]$, $[1, 0]$, $[2, 3]$ vagy $[3, 2]$ értékkel.

Példa 2

A 0. részfeladat egy további példatesztet tartalmaz $N = 256$ helyszínnel. Ez a tesztet a csatolt állományban található, amelyet a versenyrendszerből tölthetsz le.

Feltételek

- $3 \leq N \leq 256$
- Az N összege a `longest_trip` összes hívása esetén nem haladja meg az 1 024-t az egyes tesztesetekben.
- $1 \leq D \leq 3$

Részfeladatok

1. (5 pont) $D = 3$
2. (10 pont) $D = 2$
3. (25 pont) $D = 1$. Jelölje l^* a leghosszabb út hosszát. A `longest_trip` függvénynek nem kell l^* hosszúságú utat visszaadnia. Ehelyett legalább egy legalább $\left\lceil \frac{l^*}{2} \right\rceil$ hosszú utat ad vissza.
4. (60 pont) $D = 1$

A 4. részfeladatban a pontszámot a `longest_trip` egyetlen hívása során az `are_connected` függvény hívásainak száma határozza meg. Legyen q a hívások maximális száma a `longest_trip` összes hívása között a részfeladat összes tesztesetében. A részfeladat pontszáma a következő táblázat szerint számítható:

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

Ha bármelyik tesztesetben az `are_connected` függvény hívása nem felel meg a leírt feltételeknek, vagy a `longest_trip` által visszaadott tömb hibás, akkor az adott részfeladat megoldásának pontszáma 0 lesz.

Mintaértékelő

Jelölje C a tesztesetek számát, azaz a `longest_trip` hívásainak számát. A mintaértékelő a következő formátumban olvassa be a bemenetet:

- 1. sor: C

A C . teszteset leírása következik.

A mintaértékelő az egyes tesztesetek leírását a következő formátumban olvassa be:

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

Itt minden U_i ($1 \leq i < N$) egy i méretű tömb, amely azt írja le, hogy mely helyszíneket köti össze út. Minden olyan i és j esetében, ahol $1 \leq i < N$ és $0 \leq j < i$:

- ha a j és az i helyszíneket út köti össze, akkor az $U_i[j]$ értékének 1-nek kell lennie;
- ha a j és az i helyszíneket nem köti össze út, akkor a $U_i[j]$ értékének 0-nak kell lennie.

Minden egyes tesztelésben a `longest_trip` hívása előtt a mintaértékelő ellenőrzi, hogy az úthálózat sűrűsége legalább D . Ha ez a feltétel nem teljesül, akkor kiírja az `Insufficient Density` üzenetet, és megszakítja a futását.

Ha a mintaértékelő protokollsértést észlel, a mintaértékelő kimenete a `Protocol Violation: <MSG>`, ahol a `<MSG>` a következő hibaüzenetek egyike:

- `invalid array`: a `are_connected` hívásban az A és B tömbök közül legalább az egyik
 - üres, vagy
 - olyan elemet tartalmaz, amely nem 0 és $N - 1$ közötti egész szám, vagy
 - legalább kétszer ugyanazt az elemet tartalmazza.
- `non-disjoint arrays`: az `are_connected` hívásakor az A és a B tömbök nem diszjunktak.
- `too many calls`: az `are_connected` hívások száma meghaladja a 32 640-t a `longest_trip` aktuális hívása során, vagy meghaladja az 150 000-t összesen.
- `too many elements`: a `"are_connected"`-nek átadott helyszínek száma az összes hívás során meghaladja az 1 500 000-t.

Ellenkező esetben a `longest_trip` által visszaadott tömb elemei egy tesztelésben $t[0], t[1], \dots, t[l-1]$ legyenek, valamely nemnegatív l esetén. A mintaértékelő három sort ír ki ebben a tesztelésben a következő formátumban:

- 1. sor: l
- 2. sor: $t[0] \ t[1] \ \dots \ t[l-1]$
- 3. sor\$: az `are_connected` hívások száma a tesztelés során.

Végül a mintaértékelő kimenete:

- $1 + 3 \cdot C$ sor: a `are_connected` hívások maximális száma a `longest_trip` összes hívása során.