



Најдолгото патување

Организаторите на IOI 2023 се во големи проблеми! Тие заборавиле да го испланираат патувањето до Ópusztaszer за следниот ден. Можеби не е премногу доцна и можете да им помогнете ...

Имате N знаменитости во Ópusztaszer, со индекси од 0 до $N - 1$. Некои парови од овие знаменитости се поврзани со *двонасочни патеки*. Секој пар од знаменитости е поврзан со најмногу една патека. Организаторите *не знаат* кои знаменитости се поврзани со патеки.

Можеме да кажеме дека **густината** на мрежата на патеки во Ópusztaszer е **најмалку** δ ако секои 3 различни знаменитости имаат барем δ патеки помеѓу нив. Со други зборови, за секоја тројка од знаменитости (u, v, w) така што $0 \leq u < v < w < N$, помеѓу паровите знаменитости (u, v) , (v, w) и (u, w) најмалку δ парови се поврзани со патека.

Организаторите *знаат* за позитивен цел број D така што густината на мрежата на патеки е најмалку D . Да забележиме дека вредноста на D не може да биде поголема од 3.

Организаторите може да прават **повици** кон операторот на телефонската централа во Ópusztaszer за да добијат информации за патеките помеѓу одредени знаменитости. За секој повик, две низи на знаменитости $[A[0], \dots, A[P - 1]]$ и $[B[0], \dots, B[R - 1]]$ мора да бидат специфицирани. Знаменитостите треба да се различни по парови, односно,

- $A[i] \neq A[j]$ за секое i и j така што $0 \leq i < j < P$;
- $B[i] \neq B[j]$ за секое i и j така што $0 \leq i < j < R$;
- $A[i] \neq B[j]$ за секое i и j така што $0 \leq i < P$ и $0 \leq j < R$.

За секој повик, операторот дава назад информација дали има патека која ја поврзува знаменитоста од низата A со знаменитоста од низата B . Попрецизно, операторот проверува помеѓу сите парови i и j така што $0 \leq i < P$ и $0 \leq j < R$. Ако во било кој од нив, знаменитостите $A[i]$ и $B[j]$ се поврзани со патека и враќа назад true. Во спротивно, операторот дава назад false.

Патување со должина l е секвенца од *различни* знаменитости $t[0], t[1], \dots, t[l - 1]$, каде за секое i помеѓу 0 и $l - 2$, вклучително, знаменитоста $t[i]$ и знаменитоста $t[i + 1]$ се поврзани со патека. Патувањето со должина l се нарекува **најдолго патување** ако не постои ниедно друго патување со должина од најмалку $l + 1$.

Вашата задача е да им помогнете на организаторите да го најдат најдолгото патување во Ópusztaszer правејќи повици кон операторот на телефонската централа.

Имплементациски детали

Треба да ја имплементирате следната процедура:

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

- N : бројот на знаменитости во Ópusztaszer.
- D : гарантираната минимална густина на мрежата на патеки.
- Процедурата треба да врати назад низа $t = [t[0], t[1], \dots, t[l-1]]$, која ќе го претставува најдолгото патување.
- Процедурата може да биде повикана **повеќе пати** за секој тест случај.

Горната процедура може да прави повици кон следната процедура:

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

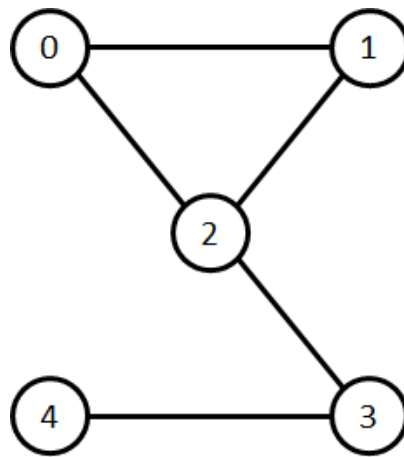
- A : непразна низа од различни знаменитости.
- B : непразна низа од различни знаменитости.
- A и B треба да се дисјунктни.
- Процедурата враќа назад `true` ако знаменитоста од A и знаменитоста од B се поврзани со патека. Во спротивно враќа `false`.
- Оваа процедура може да биде повикана најмногу 32 640 пати при секој повик на процедурата `longest_trip`, и вкупно може да биде повикана 150 000 пати.
- Вкупната должина на низите A и B дадени на оваа процедура во сите повикувања неможе да надмине 1 500 000.

Оценувачот **не е адаптивен**. Секое испратено решение се оценува на истиот сет на тест случаи. Така што, вредностите за N и D , како и паровите на знаменитости поврзани со патеки се фиксни за секој повик кон процедурата `longest_trip` во рамките на секој тест случај.

Примери

Пример 1

Земете го во предвид следното сценарио за $N = 5$, $D = 1$, и патеките помеѓу знаменитостите како на сликата:



Процедурата `longest_trip` е повикана на следниот начин:

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

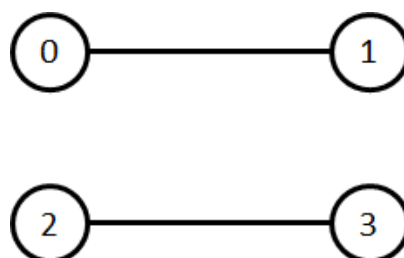
Процедурата може да прави повици со процедурата `are_connected` на следниот начин.

Повик	Парови поврзани со патека	Враќа вредност
<code>are_connected([0], [1, 2, 4, 3])</code>	(0,1) и (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>	нема парови	false

После четвртиот повик, се добива резултат *none* за паровите (1,4), (0,4), (1,3) и (0,3) се поврзани со патека. Бидејќи густината на мрежата е најмалку $D = 1$, гледаме дека тројката (0,3,4), и парот (3,4) мора да се поврзани со патека. На сличен начин, знаменитостите 0 и 1 мора да се поврзани.

Во овој момент може да се заклучи дека $t = [1, 0, 2, 3, 4]$ е патување со должина 5, и дека не постои друго патување со должина поголема од 5. Затоа, процедурата `longest_trip` може да врати резултат `[1, 0, 2, 3, 4]`.

Земете го во предвид следното сценарио за $N = 4$, $D = 1$, и патеките помеѓу знаменитостите како на сликата:



Процедурата `longest_trip` може да биде повикана на следниот начин:

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

Во ова сценарио, должината на најдолгото патување е 2. Затоа, после неколку повици на процедурата `are_connected`, процедурата `longest_trip` може да врати една од вредностите $[0, 1]$, $[1, 0]$, $[2, 3]$ или $[3, 2]$.

Пример 2

Подзадачата 0 содржи дополнителен пример со $N = 256$ знаменитости. Оваа подзадача е дел од прикачениот додаток што може да се превземе од CMS.

Ограничувања

- $3 \leq N \leq 256$
- Сумата на N за сите повикувања на процедурата `longest_trip` не надминува 1 024 за секој тест случај.
- $1 \leq D \leq 3$

Подзадачи

1. (5 поени) $D = 3$
2. (10 поени) $D = 2$
3. (25 поени) $D = 1$. Нека l^* ја означува должината на најдолгото патување. Процедурата `longest_trip` не враќа вредност назад за должина l^* . Наместо тоа, процедурата враќа вредност од должина најмалку $\left\lceil \frac{l^*}{2} \right\rceil$.
4. (60 поени) $D = 1$

Во подзадачата 4 вашите бодови се одредуваат од бројот на повикувања на процедурата `are_connected` во рамките на едно повикување на процедурата `longest_trip`. Нека q е максимумот на повикувања во сите повици на процедурата `longest_trip` во сите тест случаи на подзадачата. Вашите бодови ќе бидат пресметани според следната табела:

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

Ако, во било кој од тест случаите, повикувањата кон процедурата `are_connected` не е соодветна со ограничувањата во делот Имплементациски детали или ако низата која ја враќа `longest_trip` е погрешна, бодовите за таа подзадача ќе бидат 0.

Пример оценувач

Нека C го означува бројот на сценарија што е всушност бројот на повикувања на процедурата `longest_trip`. Пример оценувачот го чита влезот во следниот формат:

- линија 1: C

Описот на следните C сценарија следат.

Пример оценувачот го чита влезот на секое сценарио во следниот формат:

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

Тука, секој U_i ($1 \leq i < N$) е низа со големина i , која опишува кои парови на знаменитости се поврзани со патека. За секое i и j така што $1 \leq i < N$ и $0 \leq j < i$:

- ако знаменитоста j и знаменитоста i се поврзани со патека, тогаш вредноста на $U_i[j]$ е еднаква на 1;
- ако нема патека која ги поврзува знаменитоста j и знаменитоста i , тогаш вредноста на $U_i[j]$ е еднаква на 0.

Во секое сценарио, пред повикувањето на процедурата `longest_trip`, пример оценувачот проверува дали густината на мрежата на патеки е најмалку D . Доколку овој услов не е исполнет, оценувачот печати порака `Insufficient Density` и завршува.

Ако пример оценувачот детектира проблеми со комуникација, излезот на пример оценувачот ќе биде `Protocol Violation: <MSG>`, каде што `<MSG>` е една од следните пораки за грешка:

- `invalid array`: во повик на процедурата `are_connected`, ако барем една од низите A и B
 - е празна, или
 - содржи елемент кој не е цел број помеѓу 0 и $N - 1$, вклучително, или
 - содржи ист елемент барем два пати.
- `non-disjoint arrays`: во повик на процедурата `are_connected`, низите A и B не се дисјунктни.
- `too many calls`: бројот на повици направен кон процедурата `are_connected` надминува 32 640 во моменталниот повик на процедурата `longest_trip`, или надминува 150 000 вкупно.

- `too many elements`: вкупниот број на знаменитости кои се дадени на процедурата `are_connected` за сите повици надминува 1 500 000.

Во спротивно, нека елементите од низата која ја враќа процедурата `longest_trip` во некое сценарио е $t[0], t[1], \dots, t[l-1]$ за некое ненагативно l . Пример оценувачот печати 3 линии за ова сценарио во следниот формат:

- линија 1: l
- линија 2: $t[0] \ t[1] \ \dots \ t[l-1]$
- линија 3: бројот на повици на процедурата `are_connected` за ова сценарио

На крај, пример оценувачот печати излез:

- линија $1 + 3 \cdot C$: максималниот број на повици кон процедурата `are_connected` за сите повици на процедурата `longest_trip`