



أطول رحلة

لقد وقع منظمو أولمبياد المعلوماتية الدولي لعام 2023 في ورطة كبيرة، لانهم نسوا أن يخططوا للرحلة إلى أوبوستازر في اليوم التالي ولكن ربما يكونون غير متأخرين كثيراً

هناك N معلماً سياحياً (landmark) في أوبوستازر مرقمة من 0 وحتى $N - 1$ بعض أزواج المعالم السياحية متصلة مع بعضها عن طريق طرق ثنائية الاتجاه كل زوج من المعالم يتصل مع بعضه بطريق واحد على الأكثر. لا يعلم المنظمون أي من المعالم متصلة مع بعضها بطرق.

نعرف كثافة شبكة الطرق في أوبوستازر بأنها δ على الأقل إذا تحقق: من أجل كل 3 معالم مختلفة سيكون هناك على الأقل δ طريق فيما بينها. بعبارة أخرى: من أجل أي ثلاثية من المعالم (u, v, w) بحيث $0 \leq u < v < w < N$ ، من بين أزواج المعالم (u, v) , (v, w) و (u, w) يوجد δ زوج منها على الأقل متصلة بطرق.

يعرف المنظمون أن العدد الصحيح D يحقق أن كثافة شبكة الطرق هي بقيمته على الأقل. لاحظ أن قيمة D لا يمكن أن تكون أكبر من 3.

يمكن للمنظمين الاتصال هاتفياً بعامل الهاتف في أوبوستازر لجمع معلومات عن شبكة الطرق بين معالم سياحية معينة. في كل اتصال يتم تزويد العامل بمصفوفتين غير خاليتين من المعالم $[A[0], \dots, A[P - 1]]$ و $[B[0], \dots, B[R - 1]]$.

يجب على المعالم أن تكون مختلفة بشكل ثنائي كالتالي:

- $A[i] \neq A[j]$ من أجل i و j بحيث $0 \leq i < j < P$
- $B[i] \neq B[j]$ من أجل i و j بحيث $0 \leq i < j < R$
- $A[i] \neq B[j]$ من أجل i و j بحيث $0 \leq i < P$ و $0 \leq j < R$.

من أجل كل اتصال، يقوم عامل الهاتف بالإجابة فيما إذا كان يوجد طريق يصل بين أحد المعالم من المصفوفة A مع أحد المعالم من المصفوفة B ، بشكل آخر: يعيد العامل القيمة 'true' إذا كان يوجد i و j بحيث أن $0 \leq i < P$ و $0 \leq j < R$ و $A[i]$ و $B[j]$ متصلان بطريق، أما إذا لم يوجد هكذا معلمين فإنه يعيد القيمة 'false'.

الرحلة التي طولها l هي سلسلة من المعالم المختلفة $t[0], t[1], \dots, t[l - 1]$ بحيث من أجل كل i بين 0 و $l - 2$ متضمنة هذين القيمتين: المعلمان $t[i]$ و $t[i + 1]$ متصلان بطريق فيما بينهما.

تدعى الرحلة التي طولها l بأطول رحلة إذا لم يكن يوجد أي رحلة أخرى طولها على الأقل $l + 1$.

مهمتك هي مساعدة المنظمين على إيجاد أطول رحلة في أوبوستازر عن طريق إجراء عدة اتصالات مع عامل الهاتف.

تفاصيل البرمجة

يجب عليك برمجة الإجرائية التالية

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

- N : عدد المعالم السياحية في أوبوستازر.
- D : الكثافة الدنيا لشبكة الطرق المعروفة من قبل المنظمين .
- هذه الإجرائية يجب أن تعيد المصفوفة $t = [t[0], t[1], \dots, t[l-1]]$, التي تمثل أطول رحلة..
- يمكن أن يتم استدعاء هذه الإجرائية عدة مرات في كل حالة اختبار.

يمكن للإجرائية السابقة أن تقوم باستدعاء الإجرائية التالية:

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

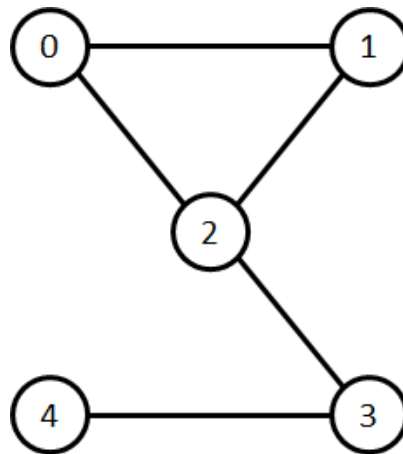
- A : مصفوفة غير خالية من المعالم المختلفة.
- B : مجموعة غير خالية من المعالم المختلفة
- A و B يجب أن تكون منفصلة عن بعضها..
- تعيد هذه الإجرائية القيمة true إذا كان أحد المعالم من A وأحد المعالم من B متصلين بطريق. وإلا تعيد القيمة false.
- يمكن استدعاء هذه الإجرائية 32 640 مرة في كل استدعاء للتابع longest_trip, وعلى الأكثر 150 000 مرة في المجموع الكلي.
- المجموع الكلي لطول المصفوفتين A و B التي سيتم تمريرها لهذه الإجرائية في كل الاستدعاءات لا يجب أن يتجاوز 1 500 000.

المقيم لن يكون متكيفاً أي أن قيم N و D بالإضافة إلى أزواج المعالم المتصلة بطرق ثابتة ومحددة قبل استدعاء التابع longest_trip.

أمثلة

المثال 1

ليكن لدينا السيناريو التالي بحيث $N = 5$, $D = 1$, ووصلات الطرق كما هو موضح في الشكل التالي



يتم استدعاء الإجرائية longest_trip بالشكل التالي:

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

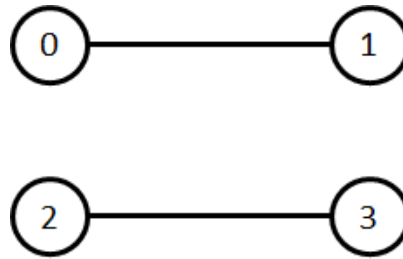
يمكن أن تقوم الإجرائية باستدعاء are_connected كالتالي.

الاستدعاء	الأزواج المتصلة بطرق	القيمة المعادة
are_connected([0], [1, 2, 4, 3])	(0,1) and (0,2)	true
are_connected([2], [0])	(2,0)	true
are_connected([2], [3])	(2,3)	true
are_connected([1, 0], [4, 3])	none	false

بعد الاستدعاء الرابع، سيكون واضحاً أن لا يوجد أي زوج من الأزواج (0,4), (1,4), (0,3), (1,3) و (0,3) متصلة بطريق. وبما أن كثافة شبكة الطرق هي على الأقل $D = 1$ ، يمكننا أن نستنتج أنه من الثلاثية (0,3,4)، الزوج (3,4) يجب أن يكون متصلاً بطريق.

في هذه النقطة، يمكن استنتاج أن $t = [1, 0, 2, 3, 4]$ هي رحلة طولها 5 وأنه لا يوجد أي رحلة أخرى طولها أكبر من 5

ليكن السيناريو الآخر بحيث $N = 4, D = 1$ ، والطرق بين المعالم هي كما هو موضح في الشكل التالي:



يتم استدعاء الإجرائية longest_trip بالطريقة التالية:

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

في هذا السيناريو، طول أطول رحلة هو 2، لذلك بعد عدة استدعاءات للإجرائية are_connected، سيقوم الإجراء longest_trip بإعادة أي مما يلي [0,1], [1,0], [2,3] أو [3,2].

محاثال 2

المسألة الجزئية 0 تحوي حالة اختبار كمثال يكون فيها عدد المعالم السياحية $N = 256$ (landmarks). حالة الاختبار هذه مضمنة ضمن الحزمة المرفقة والتي يمكنك تحميلها من نظام الاختبارات.

الحدود

- $3 \leq N \leq 256$
- مجموع كل قيم N في كل استدعاءات longest_trip لن تتجاوز 1.024.

$$1 \leq D \leq 3 \quad \bullet$$

المسائل الجزئية

1. (5 درجات) $D = 3$

2. (10 درجات) $D = 2$

3. (25 درجات) $D = 1$. ليكن l^* يمثل طول أطول رحلة. الإجرائية longest_trip ليس بالضرورة أن تعيد رحلة طولها l^* . وإنما من الممكن أن تعيد رحلة طولها على الأقل $\left\lceil \frac{l^*}{2} \right\rceil$.

4. (60 درجة) $D = 1$

إذا تحقق أنه في أحد حالات الاختبار لم تتوافق استدعاءات التابع are_connected مع الحدود المشروحة في تفاصيل البرمجة، أو إذا كانت المصفوفة التي أعادها التابع longest_trip غير صحيحة، سينال حلك لهذه المسألة الجزئية علامة 0.

في المسألة الجزئية 4 يتم تحديد علامتك بحسب عدد مرات استدعاء الإجرائية are_connected في كل استدعاء للإجرائية longest_trip. ليكن q هو أكبر عدد من الاستدعاءات في كل استدعاءات الإجرائية longest_trip في كل حالة اختبار من المسألة الجزئية ستكون علامتك على هذه المسألة الجزئية محسوبة وفق الجدول التالي:

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

Sample Grader

Let C denote the number of scenarios, that is, the number of calls to longest_trip. The sample grader reads the input in the following format:

C : 1 line \bullet

The descriptions of C scenarios follow

The sample grader reads the description of each scenario in the following format

N D : 1 line \bullet

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

Here, each U_i ($1 \leq i < N$) is an array of size i , describing which pairs of landmarks are connected by a road. For each i and j such that $1 \leq i < N$ and $0 \leq j < i$

- \bullet if landmarks j and i are connected by a road, then the value of $U_i[j]$ should be 1;
- \bullet if there is no road connecting landmarks j and i , then the value of $U_i[j]$ should be 0.

In each scenario, before calling `longest_trip`, the sample grader checks whether the density of the road network is at least D . If this condition is not met, it prints the message `Insufficient`
`.Density` and terminates

If the sample grader detects a protocol violation, the output of the sample grader is `Protocol`
`:Violation: <MSG>`, where `<MSG>` is one of the following error messages

- `B` invalid array: in a call to `are_connected`, at least one of arrays A and B
 - is empty, or
 - contains an element that is not an integer between 0 and $N - 1$, inclusive, or
 - contains the same element at least twice
- `.non-disjoint` arrays: in a call to `are_connected`, arrays A and B are not disjoint
- `too many calls`: the number of calls made to `are_connected` exceeds 32 640 over the
`.current` invocation of `longest_trip`, or exceeds 150 000 in total
- `too many elements`: the total number of landmarks passed to `are_connected` over all
`.1 500 000` calls exceeds

Otherwise, let the elements of the array returned by `longest_trip` in a scenario be $t[0], t[1], \dots, t[l - 1]$ for some nonnegative l . The sample grader prints three lines for this scenario
in the following format

- l :1 line
- $t[0] \ t[1] \ \dots \ t[l - 1]$:2 line
- line 3: the number of calls to `are_connected` over this scenario

Finally, the sample grader prints

- line $1 + 3 \cdot C$: the maximum number of calls to `are_connected` over all calls to
`longest_trip`