



Longest Trip

Các nhà tổ chức IOI 2023 đang gặp khó khăn lớn! Họ đã quên kế hoạch chuyến đi đến Ópusztaszer cho ngày mai. Nhưng có lẽ chưa quá muộn...

Ở Ópusztaszer có N điểm du lịch được đánh số từ 0 đến $N - 1$. Một số cặp của những điểm này được nối bởi **các con đường hai chiều**. Mỗi cặp điểm du lịch được nối bởi tối đa một con đường. Các nhà tổ chức *không biết* những điểm nào được nối bởi các con đường.

Chúng ta nói rằng **mật độ** của mạng lưới đường đi ở Ópusztaszer **tối thiểu** là δ nếu mỗi bộ 3 điểm du lịch khác nhau có ít nhất δ con đường giữa chúng. Nói cách khác, đối với mỗi bộ 3 điểm (u, v, w) sao cho $0 \leq u < v < w < N$, trong số các cặp điểm (u, v) , (v, w) và (u, w) ít nhất δ cặp được nối bởi một con đường.

Các nhà tổ chức *biết* một số nguyên dương D sao cho mật độ của mạng lưới đường đi tối thiểu là D . Lưu ý rằng giá trị của D không thể lớn hơn 3.

Các nhà tổ chức có thể thực hiện **các cuộc gọi** tới bộ phận điều phối ở Ópusztaszer để thu thập thông tin về các kết nối đường đi giữa một vài điểm du lịch. Trong mỗi cuộc gọi, họ cần xác định hai mảng không rỗng của các điểm du lịch $[A[0], \dots, A[P - 1]]$ và $[B[0], \dots, B[R - 1]]$. Các điểm du lịch phải là các cặp đôi một khác nhau, nghĩa là,

- $A[i] \neq A[j]$ với mọi i và j sao cho $0 \leq i < j < P$;
- $B[i] \neq B[j]$ với mọi i và j sao cho $0 \leq i < j < R$;
- $A[i] \neq B[j]$ với mọi i và j sao cho $0 \leq i < P$ và $0 \leq j < R$.

Đối với mỗi cuộc gọi, bộ phận điều phối sẽ báo cáo xem có con đường nối giữa một điểm du lịch từ A và một điểm du lịch từ B hay không. Cụ thể, bộ phận điều phối xét qua tất cả các cặp i và j sao cho $0 \leq i < P$ và $0 \leq j < R$. Nếu đối với bất kỳ cặp nào trong chúng, các điểm du lịch $A[i]$ và $B[j]$ được kết nối bởi một con đường, bộ phận điều phối trả về `true`. Nếu không, bộ phận điều phối trả về `false`.

Một **hành trình** có độ dài l là một chuỗi các điểm du lịch *khác nhau* $t[0], t[1], \dots, t[l - 1]$, trong đó đối với mỗi i từ 0 đến $l - 2$, kể cả 0 và $l - 2$, điểm $t[i]$ và điểm $t[i + 1]$ được nối bởi một con đường. Một chuyến đi có độ dài l được gọi là một **hành trình dài nhất** nếu không tồn tại bất kỳ hành trình nào có độ dài tối thiểu là $l + 1$.

Nhiệm vụ của bạn là giúp các nhà tổ chức tìm một hành trình dài nhất ở Ópusztaszer bằng cách thực hiện các cuộc gọi tới bộ phận điều phối.

Chi tiết cài đặt

Bạn cần cài đặt hàm sau:

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

- N : số lượng các điểm du lịch ở Ópusztaszer.
- D : mật độ tối thiểu được đảm bảo của mạng lưới đường đi.
- Hàm này cần trả về một mảng $t = [t[0], t[1], \dots, t[l - 1]]$, biểu diễn một hành trình dài nhất.
- Hàm này có thể được gọi **hiều lần** trong mỗi test.

Hàm trên có thể gọi đến hàm sau:

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

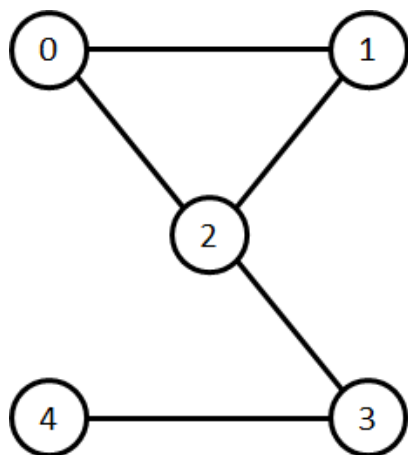
- A : một mảng không rỗng của các điểm du lịch khác nhau.
- B : một mảng không rỗng của các điểm du lịch khác nhau.
- A và B phải không có phần tử chung.
- Hàm này trả về true nếu có một điểm du lịch từ A và một điểm du lịch từ B được nối bởi một con đường. Ngược lại, hàm trả về false.
- Hàm này có thể được gọi tối đa 32 640 lần trong mỗi lời gọi longest_trip, và tối đa 150 000 lần tổng cộng.
- Tổng chiều dài của các mảng A và B được truyền vào hàm này qua tất cả các lần gọi không vượt quá 1, 500, 000.

Trình chấm **không thích nghi**. Mỗi bài nộp được chấm điểm trên cùng một bộ test. Nghĩa là, các giá trị của N và D , cũng như các cặp điểm du lịch được kết nối bởi các con đường, đều được cố định đối với mỗi lời gọi longest_trip trong một test.

Các ví dụ

Ví dụ 1

Xét kịch bản khi $N = 5$, $D = 1$, và các kết nối đường đi được chỉ ra trong hình vẽ sau:



Hàm `longest_trip` được gọi theo cách sau:

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

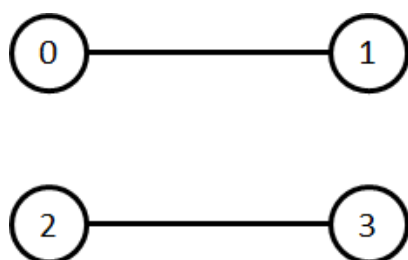
Hàm này có thể thực hiện các lời gọi đến hàm `are_connected` như sau.

Lời gọi	Các cặp được nối bởi một con đường	Giá trị trả về
<code>are_connected([0], [1, 2, 4, 3])</code>	(0, 1) và (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>	không có	false

Sau lần gọi thứ tư, nhận thấy *không có* cặp nào trong số các cặp (1, 4), (0, 4), (1, 3) và (0, 3) được kết nối bởi một con đường. Vì mật độ của mạng ít nhất là $D = 1$, ta thấy rằng từ bộ ba (0, 3, 4), cặp (3, 4) phải được kết nối bằng một con đường. Tương tự như vậy, các điểm du lịch 0 và 1 phải được kết nối.

Tại thời điểm này, có thể kết luận rằng $t = [1, 0, 2, 3, 4]$ là một hành trình có độ dài 5 và không tồn tại một hành trình có độ dài lớn hơn 5. Do đó, hàm `longest_trip` có thể trả về `[1, 0, 2, 3, 4]`.

Xét một kịch bản khác khi $N = 4$, $D = 1$, và các con đường giữa các điểm du lịch được chỉ ra trong hình sau:



Hàm `longest_trip` được gọi theo cách sau:

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

Trong kịch bản này, độ dài của hành trình dài nhất là 2. Do đó, sau một vài lời gọi đến hàm `are_connected`, hàm `longest_trip` có thể trả về một trong số $[0, 1]$, $[1, 0]$, $[2, 3]$ or $[3, 2]$.

Ví dụ 2

Subtask 0 chứa thêm một test với $N = 256$ điểm du lịch. Test này có trong gói đính kèm mà bạn có thể tải về từ hệ thống cuộc thi.

Các ràng buộc

- $3 \leq N \leq 256$
- Tổng của tất cả giá trị N trong tất cả lời gọi đến `longest_trip` không vượt quá 1 024 trong một test.
- $1 \leq D \leq 3$

Các subtask

1. (5 điểm) $D = 3$
2. (10 điểm) $D = 2$
3. (25 điểm) $D = 1$. Gọi l^* là chiều dài của hành trình dài nhất. Hàm `longest_trip` không phải trả về một hành trình có độ dài l^* . Thay vào đó, hàm cần trả về một hành trình có độ dài tối thiểu là $\left\lceil \frac{l^*}{2} \right\rceil$.
4. (60 điểm) $D = 1$

Trong subtask 4, điểm của bạn được xác định dựa trên số lần gọi hàm `are_connected` qua một lệnh gọi `longest_trip`. Gọi q là số lượng cuộc gọi tối đa trong số tất cả các lời gọi `longest_trip` trong mọi test của subtask. Điểm của bạn cho subtask này được tính theo bảng sau:

Điều kiện	Điểm
$2\,750 < q \leq 32\,640$	20
$550 < q \leq 2\,750$	30
$400 < q \leq 550$	45
$q \leq 400$	60

Nếu trong bất kỳ test nào các lệnh gọi hàm `are_connected` không tuân theo các ràng buộc được mô tả trong phần Chi tiết cài đặt, hoặc mảng được trả về bởi `longest_trip` không đúng, điểm cho lời giải của bạn đối với subtask tương ứng sẽ là 0.

Trình chấm mẫu

Gọi C là số lượng các kịch bản, nghĩa là, số lượng các lời gọi đến `longest_trip`. Trình chấm mẫu đọc dữ liệu vào theo mẫu sau:

- dòng 1: C

Theo sau là các mô tả của C kịch bản.

Trình chấm mẫu đọc mô tả mỗi kịch bản theo mẫu sau:

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

Ở đây, mỗi U_i ($1 \leq i < N$) là một mảng kích thước i , mô tả các cặp điểm du lịch nào được nối bởi một con đường. Với mỗi i và j sao cho $1 \leq i < N$ và $0 \leq j < i$:

- nếu các điểm j và i được nối bởi một con đường, thì giá trị của $U_i[j]$ phải là 1;
- nếu không có con đường nào nối các điểm du lịch j và i , thì giá trị của $U_i[j]$ phải là 0.

Trong mỗi kịch bản, trước khi gọi `longest_trip`, trình chấm mẫu sẽ kiểm tra xem mật độ của mạng lưới đường đi có ít nhất là D hay không. Nếu điều kiện này không thoả mãn, nó sẽ in thông báo `Insufficient Density` và kết thúc.

Nếu trình chấm mẫu phát hiện vi phạm giao thức, đầu ra của trình chấm mẫu là `Protocol Violation: <MSG>`, trong đó `<MSG>` là một trong các thông báo lỗi sau:

- `invalid array`: trong một lời gọi tới `are_connected`, ít nhất một trong hai mảng A và B
 - là rỗng, hoặc
 - chứa một phần tử mà không là một số nguyên nằm giữa 0 và $N - 1$, kể cả 0 và $N - 1$, hoặc
 - có một phần tử xuất hiện ít nhất hai lần.
- `non-disjoint arrays`: trong một lời gọi tới `are_connected`, các mảng A và B là có phần tử chung.
- `too many calls`: số lượng các lời gọi đến `are_connected` vượt quá 32 640 qua lời gọi hiện tại tới `longest_trip`, hoặc vượt quá 150 000 tổng cộng.
- `too many elements`: tổng cộng số lượng các điểm du lịch được truyền vào `are_connected` trên tất cả các lời gọi vượt quá 1 500 000.

Ngược lại, gọi các phần tử của mảng được trả về bởi `longest_trip` trong một kịch bản là $t[0], t[1], \dots, t[l - 1]$ đối với một số giá trị không âm l . Trình chấm mẫu in ra ba dòng đối với kịch bản này theo định dạng sau:

- dòng 1: l
- dòng 2: $t[0] t[1] \dots t[l - 1]$
- dòng 3: số lượng các lời gọi tới `are_connected` qua kịch bản này

Cuối cùng, trình chấm mẫu in:

- dòng $1 + 3 \cdot C$: số lượng các lời gọi tới `are_connected` qua tất cả các lời gọi tới `longest_trip`