

Keys

Kiến trúc sư Timothy đã thiết kế một trò chơi thoát hiểm mới. Trong trò chơi này, có n phòng được đánh số từ 0 đến $n - 1$. Ban đầu, mỗi phòng có đúng một chìa khóa. Mỗi khóa thuộc một loại, là một số nguyên trong khoảng từ 0 đến $n - 1$, bao gồm cả hai đầu mút. Loại chìa khóa trong phòng i ($0 \leq i \leq n - 1$) là $r[i]$. Lưu ý rằng nhiều phòng có thể có chìa khóa cùng loại, tức là các giá trị $r[i]$ không nhất thiết phải khác nhau.

Ngoài ra còn có m đường kết nối **hai chiều** trong trò chơi, được đánh số từ 0 đến $m - 1$. Đường kết nối j ($0 \leq j \leq m - 1$) liên kết một cặp phòng khác nhau $u[j]$ và $v[j]$. Một cặp phòng có thể được liên kết bằng nhiều đường kết nối.

Trò chơi được chơi bởi một người chơi thu thập các chìa khóa và di chuyển giữa các phòng bằng cách đi qua các đường kết nối. Chúng ta nói rằng người chơi **đi qua** đường kết nối j khi họ sử dụng đường kết nối này để di chuyển từ phòng $u[j]$ sang phòng $v[j]$ hoặc ngược lại. Người chơi chỉ có thể đi qua đường kết nối j nếu trước đó họ đã thu thập được chìa khóa loại $c[j]$.

Tại bất kỳ thời điểm nào trong trò chơi, người chơi đang ở trong một căn phòng x nào đó và có thể thực hiện hai loại hành động:

- thu thập chìa khóa trong phòng x , là loại chìa khóa $r[x]$ (trừ khi họ đã thu thập nó rồi),
- đi qua đường kết nối j , trong đó $u[j] = x$ hoặc $v[j] = x$, nếu trước đó người chơi đã thu thập một khóa loại $c[j]$. Lưu ý rằng người chơi **không bao giờ** bỏ đi chìa khóa mà họ đã thu thập.

Người chơi **bắt đầu** trò chơi ở phòng s và không mang theo bất kỳ chìa khóa nào. Phòng t **có thể đến được** từ phòng s , nếu người chơi bắt đầu trò chơi ở phòng s có thể thực hiện một dãy hành động được mô tả ở trên và đến phòng t .

Đối với mỗi phòng i ($0 \leq i \leq n - 1$), kí hiệu số lượng phòng có thể đến được từ phòng i là $p[i]$. Timothy muốn biết tập hợp các chỉ số i đạt được giá trị nhỏ nhất trong các $p[i]$ với $0 \leq i \leq n - 1$.

Chi tiết cài đặt

Bạn cần cài đặt hàm sau:

```
int[] find_reachable(int[] r, int[] u, int[] v, int[] c)
```

- r : một mảng độ dài n . Với mỗi i ($0 \leq i \leq n - 1$), chìa khóa trong phòng i là loại $r[i]$.
- u, v : hai mảng độ dài m . Với mỗi j ($0 \leq j \leq m - 1$), đường kết nối j liên kết các phòng $u[j]$ và $v[j]$.

- c : một mảng độ dài m . Với mỗi j ($0 \leq j \leq m - 1$), loại chìa khóa cần để đi qua đường kết nối j là $c[j]$.
- Hàm này cần trả về một mảng a độ dài n . Với mỗi $0 \leq i \leq n - 1$, giá trị $a[i]$ là 1 nếu với mọi j thỏa mãn $0 \leq j \leq n - 1$, $p[i] \leq p[j]$. Ngược lại, giá trị $a[i]$ là 0.

Các ví dụ

Ví dụ 1

Xem xét lời gọi dưới đây:

```
find_reachable([0, 1, 1, 2],
               [0, 0, 1, 1, 3], [1, 2, 2, 3, 1], [0, 0, 1, 0, 2])
```

Nếu người chơi bắt đầu trò chơi ở phòng 0, họ có thể thực hiện dãy các hành động sau:

Phòng hiện tại	Hành động
0	Thu thập chìa khóa loại 0
0	Đi qua đường kết nối 0 đến phòng 1
1	Thu thập chìa khóa loại 1
1	Đi qua đường kết nối 2 đến phòng 2
2	Đi qua đường kết nối 2 đến phòng 1
1	Đi qua đường kết nối 3 đến phòng 3

Do đó, phòng 3 có thể đến được từ phòng 0. Tương tự, chúng ta có thể xây dựng các dãy hành động cho thấy tất cả các phòng đều có thể đến được từ phòng 0, tức là $p[0] = 4$. Bảng dưới đây cho biết các phòng có thể đến được với từng phòng xuất phát:

Phòng xuất phát i	Các phòng đến được	$p[i]$
0	[0, 1, 2, 3]	4
1	[1, 2]	2
2	[1, 2]	2
3	[1, 2, 3]	3

Giá trị nhỏ nhất của $p[i]$ trên tất cả các phòng là 2 và giá trị này đạt được với $i = 1$ hoặc $i = 2$. Do đó, hàm này sẽ trả về [0, 1, 1, 0].

Ví dụ 2

```
find_reachable([0, 1, 1, 2, 2, 1, 2],
               [0, 0, 1, 1, 2, 3, 3, 4, 4, 5],
               [1, 2, 2, 3, 3, 4, 5, 5, 6, 6],
               [0, 0, 1, 0, 0, 1, 2, 0, 2, 1])
```

Bảng dưới đây cho biết các phòng có thể đến được:

Phòng xuất phát i	Các phòng đến được	$p[i]$
0	[0, 1, 2, 3, 4, 5, 6]	7
1	[1, 2]	2
2	[1, 2]	2
3	[3, 4, 5, 6]	4
4	[4, 6]	2
5	[3, 4, 5, 6]	4
6	[4, 6]	2

Giá trị nhỏ nhất của $p[i]$ trên tất cả các phòng là 2 và giá trị này đạt được với $i \in \{1, 2, 4, 6\}$. Do đó, hàm này sẽ trả về [0, 1, 1, 0, 1, 0, 1].

Ví dụ 3

```
find_reachable([0, 0, 0], [0], [1], [0])
```

Bảng dưới đây cho biết các phòng có thể đến được:

Phòng xuất phát i	Các phòng tới được	$p[i]$
0	[0, 1]	2
1	[0, 1]	2
2	[2]	1

Giá trị nhỏ nhất của $p[i]$ trên tất cả các phòng là 1 và giá trị này đạt được khi $i = 2$. Do đó, hàm này sẽ trả về [0, 0, 1].

Các ràng buộc

- $2 \leq n \leq 300\,000$
- $1 \leq m \leq 300\,000$
- $0 \leq r[i] \leq n - 1$ với mọi $0 \leq i \leq n - 1$
- $0 \leq u[j], v[j] \leq n - 1$ và $u[j] \neq v[j]$ với mọi $0 \leq j \leq m - 1$

- $0 \leq c[j] \leq n - 1$ với mọi $0 \leq j \leq m - 1$

Các subtask

1. (9 điểm) $c[j] = 0$ với mọi $0 \leq j \leq m - 1$ và $n, m \leq 200$
2. (11 điểm) $n, m \leq 200$
3. (17 điểm) $n, m \leq 2000$
4. (30 điểm) $c[j] \leq 29$ (với mọi $0 \leq j \leq m - 1$) và $r[i] \leq 29$ (với mọi $0 \leq i \leq n - 1$)
5. (33 điểm) Không có ràng buộc nào thêm.

Trình chấm mẫu

Trình chấm mẫu đọc dữ liệu vào theo định dạng sau:

- dòng 1: $n \ m$
- dòng 2: $r[0] \ r[1] \ \dots \ r[n - 1]$
- dòng $3 + j$ ($0 \leq j \leq m - 1$): $u[j] \ v[j] \ c[j]$

Trình chấm mẫu ghi giá trị trả về của hàm `find_reachable` theo định sau:

- dòng 1: $a[0] \ a[1] \ \dots \ a[n - 1]$