

Kunci

Timothy, sang arsitek, telah merancang sebuah permainan *escape game* baru. Pada game ini, terdapat n buah ruangan yang dinomori dari 0 sampai $n - 1$. Awalnya, setiap ruangan memiliki tepat satu buah kunci. Setiap kunci memiliki sebuah tipe, yang merupakan sebuah bilangan bulat diantara 0 dan $n - 1$, inklusif. Tipe kunci pada ruangan i ($0 \leq i \leq n - 1$) adalah $r[i]$. Perlu dicatat bahwa beberapa ruangan bisa saja memiliki kunci dengan tipe yang sama, dengan kata lain, nilai-nilai $r[i]$ tidak harus berbeda.

Terdapat juga m buah penghubung **dua arah** pada permainan ini, dinomori dari 0 sampai $m - 1$. Penghubung j ($0 \leq j \leq m - 1$) menghubungkan 2 buah ruangan berbeda $u[j]$ dan $v[j]$. 2 buah ruangan dapat dihubungkan oleh beberapa penghubung.

Permainan ini dimainkan oleh seorang pemain yang mengumpulkan kunci dan berpindah-pindah antar ruangan melalui penghubung-penghubung ruangan. Seorang pemain dikatakan **melalui** sebuah penghubung j apabila dia menggunakan penghubung tersebut untuk berpindah dari ruangan $u[j]$ ke ruangan $v[j]$, ataupun sebaliknya. Sang pemain hanya dapat melalui penghubung j apabila dia telah mengumpulkan sebuah kunci dengan tipe $c[j]$ sebelumnya.

Pada setiap saat dalam permainan, sang pemain akan berada pada suatu ruangan x dan dapat melakukan 2 jenis hal:

- Mengambil kunci ruangan x , yang tipenya adalah $r[x]$ (kecuali sang pemain telah mengumpulkannya sebelumnya).
- Melalui sebuah penghubung j , dimana $u[j] = x$ atau $v[j] = x$, apabila sang pemain telah mengumpulkan kunci $c[j]$ sebelumnya. Perlu dicatat bahwa sang pemain **tidak pernah** membuang kunci yang telah ia kumpulkan sebelumnya.

Sang pemain **memulai** permainan ini pada sebuah ruangan s tanpa membawa kunci apapun. Sebuah ruangan t dikatakan **terjangkau** dari sebuah ruangan s , apabila sang pemain yang memulai permainan dari ruangan s dapat melakukan serangkaian tindakan, seperti yang telah dijelaskan diatas, dan mencapai ruangan t .

Untuk setiap ruangan i ($0 \leq i \leq n - 1$), nyatakanlah jumlah ruangan yang terjangkau dari ruangan i sebagai $p[i]$. Timothy ingin mengetahui himpunan indeks i yang memiliki nilai $p[i]$ terkecil dari seluruh $0 \leq i \leq n - 1$.

Detail Implementasi

Anda harus mengimplementasi fungsi berikut:

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

- r : sebuah array dengan panjang n . Untuk setiap i ($0 \leq i \leq n - 1$), kunci pada ruangan i memiliki tipe $r[i]$.
- u, v : 2 buah array dengan panjang m . Untuk setiap j ($0 \leq j \leq m - 1$), penghubung j menghubungkan ruangan $u[j]$ dan $v[j]$.
- c : sebuah array dengan panjang m . Untuk setiap j ($0 \leq j \leq m - 1$), tipe kunci yang dibutuhkan untuk melalui penghubung j adalah $c[j]$.
- Fungsi ini harus mengembalikan sebuah array a dengan panjang n . Untuk setiap $0 \leq i \leq n - 1$, nilai $s[i]$ merupakan 1 untuk setiap j dimana $0 \leq j \leq n - 1$, $p[i] \leq p[j]$. Selain itu, nilai $s[i]$ merupakan 0.

Contoh

Contoh 1

Perhatikan pemanggilan berikut:

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

Jika sang pemain memulai permainan pada ruangan 0, mereka bisa melakukan serangkaian tindakan berikut:

Ruangan sekarang	Tindakan
0	Mengambil kunci dengan tipe 0
0	Melalui penghubung 0 menuju ruangan 1
1	Mengambil kunci dengan tipe 1
1	Melalui penghubung 2 menuju ruangan 2
2	Melalui penghubung 2 menuju ruangan 1
1	Melalui penghubung 3 menuju ruangan 3

Oleh karena itu, ruangan 3 terjangkau dari ruangan 0. Dengan cara yang serupa, kita dapat membuat serangkaian tindakan yang menunjukkan bahwa semua ruangan terjangkau dari ruangan 0, yang berarti $p[0] = 4$. Tabel dibawah ini menunjukkan daftar ruangan yang terjangkau untuk setiap ruangan awal:

Ruangan awal i	Ruangan yang terjangkau	$p[i]$
0	[0, 1, 2, 3]	4
1	[1, 2]	2
2	[1, 2]	2
3	[1, 2, 3]	3

Nilai $p[i]$ terkecil dari seluruh ruangan adalah 2, dan nilai ini dicapai oleh $i = 1$ or $i = 2$. Oleh karena itu, fungsi ini harus mengembalikan [0, 1, 1, 0].

Contoh 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])
```

Tabel berikut ini menunjukkan daftar ruangan yang terjangkau:

Ruangan awal i	Ruangan yang terjangkau	$p[i]$
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

Nilai $p[i]$ terkecil dari seluruh ruangan adalah 2, dan nilai ini dicapai oleh $i \in \{1, 2, 4, 6\}$. Oleh karena itu, fungsi ini harus mengembalikan [0, 1, 1, 0, 1, 0, 1].

Contoh 3

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

Tabel berikut ini menunjukkan daftar ruangan yang terjangkau:

Ruangan awal i	Ruangan yang terjangkau	$p[i]$
0	$[0, 1]$	2
1	$[0, 1]$	2
2	$[2]$	1

Nilai $p[i]$ terkecil dari seluruh ruangan adalah 1, dan nilai ini dicapai oleh $i = 2$. Oleh karena itu, fungsi ini harus mengembalikan $[0, 0, 1]$.

Batasan

- $2 \leq n \leq 300\,000$
- $1 \leq m \leq 300\,000$
- $0 \leq r[i] \leq n - 1$ untuk semua $0 \leq i \leq n - 1$
- $0 \leq u[j], v[j] \leq n - 1$ dan $u[j] \neq v[j]$ untuk semua $0 \leq j \leq m - 1$
- $0 \leq c[j] \leq n - 1$ untuk semua $0 \leq j \leq m - 1$

Subsoal

1. (9 poin) $c[j] = 0$ untuk semua $0 \leq j \leq m - 1$ dan $n, m \leq 200$
2. (11 poin) $n, m \leq 200$
3. (17 poin) $n, m \leq 2000$
4. (30 poin) $c[j] \leq 29$ (untuk semua $0 \leq j \leq m - 1$) dan $r[i] \leq 29$ (untuk semua $0 \leq i \leq n - 1$)
5. (33 poin) Tidak ada batasan tambahan.

Contoh Grader

Contoh *grader* membaca input dengan format sebagai berikut:

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

Contoh *grader* mencetak nilai yang dikembalikan oleh `find_reachable` dengan format sebagai berikut:

- baris 1: $s[0] \ s[1] \ \dots \ s[n - 1]$