

Keys

Arhitekta Tim osmislio je novu igru bježanja. U ovoj igri imamo n soba označenih brojevima od 0 do $n - 1$. U početku svaka soba sadrži tačno jedan ključ. Svaki ključ ima tip, koji je cijeli broj između 0 i $n - 1$, uključivo. Tip ključa u sobi i ($0 \leq i \leq n - 1$) je $r[i]$. Imajte na umu da više soba može sadržavati ključeve istog tipa, tj. vrijednosti $r[i]$ ne moraju biti različite.

Postoji takođe i m **dvosmjernih** konektora, označenih brojevima od 0 do $m - 1$. Konektor j ($0 \leq j \leq m - 1$) povezuje par različitih soba $u[j]$ i $v[j]$. Par soba može biti povezan sa više od jednog konektora.

Igru igra jedan igrač koji sakuplja ključeve i kreće se od sobe do sobe koristeći date konektore. Kažemo da igrač **prelazi** konektor j kada koristi ovaj konektor za prelazak iz sobe $u[j]$ u sobu $v[j]$, ili obrnuto. Igrač može preći konektor j samo ako je ranije prikupio ključ tipa $c[j]$.

U bilo kojem trenutku igre, igrač se nalazi u nekoj sobi x i može izvesti jednu od dvije radnje:

- sakupiti ključ u sobi x , čiji je tip $r[x]$ (osim ako ga već nije ranije sakupio),
- preći konektor j , pri čemu je $u[j] = x$ ili $v[j] = x$, pod uslovom da je igrač prethodno prikupio ključ tipa $c[j]$. Imajte na umu da igrač **nikad** ne odbacuje ključ koji je već sakupio.

Igrač **započinje** igru u nekoj sobi s i na početku nema ni jedan ključ. Soba t je **dostupna** iz sobe s , ako igrač koji započne igru u sobi s može izvršiti neki slijed gore opisanih radnji i doći do sobe t .

Za svaku sobu i ($0 \leq i \leq n - 1$), označimo broj soba do kojih se može doći iz sobe i sa $p[i]$. Timothy bi želio znati skup indeksa i koji postižu minimalnu vrijednost $p[i]$ po svim i , $0 \leq i \leq n - 1$.

Detalji implementacije

Trebate implementirati sljedeći proceduru:

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

- r : niz dužine n . Za svaki i ($0 \leq i \leq n - 1$), ključ u sobi i je tipa $r[i]$.
- u, v : dva niza dužine m . Za svaki j ($0 \leq j \leq m - 1$), konektor j povezuje sobe $u[j]$ i $v[j]$.
- c : niz dužine m . Za svaki j ($0 \leq j \leq m - 1$), tip ključa potreban za prelazak konektora j je $c[j]$.
- Ova procedura treba da vrati niz a dužine n . Za svaki $0 \leq i \leq n - 1$, vrijednost $a[i]$ treba biti 1 ako za svaki j takav da $0 \leq j \leq n - 1$, $p[i] \leq p[j]$. Inače, vrijednost $a[i]$ treba da je 0 .

Primjeri

Primjer 1

Razmotrite sljedeći poziv:

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

Ako igrač započne igru u sobi 0, može izvršiti sljedeći niz radnji:

Trenutna soba	Akcija
0	Pokupi ključ tipa 0
0	Prelazi konektor 0 do sobe 1
1	Pokupi ključ tipa 1
1	Prelazi konektor 2 do sobe 2
2	Prelazi konektor 2 do sobe 1
1	Prelazi konektor 3 do sobe 3

Dakle, soba 3 je dostupna počevši iz sobe 0. Na sličan način možemo konstruisati nizove koji pokazuju da su sve sobe dostupne iz sobe 0, što onda znači da je $p[0] = 4$. Tabela u nastavku prikazuje dostupne prostorije za sve početne sobe:

Početna soba i	Dostupne sobe	$p[i]$
0	[0, 1, 2, 3]	4
1	[1, 2]	2
2	[1, 2]	2
3	[1, 2, 3]	3

Najmanja vrijednost of $p[i]$ po svim sobama je 2, a to se postiže za $i = 1$ or $i = 2$. Stoga bi ova procedura trebala vratiti [0, 1, 1, 0].

Primjer 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])
```

Tabela ispod prikazuje dostupne sobe:

Početna soba i	Dostupne sobe	$p[i]$
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

Najmanja vrijednost $p[i]$ po svim sobama je 2, a to se postiže za $i \in \{1, 2, 4, 6\}$. Stoga bi ova procedura trebala vratiti [0, 1, 1, 0, 1, 0, 1].

Primjer 3

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

Tabela ispod prikazuje dostupne sobe:

Početna soba i	Dostupne sobe	$p[i]$
0	[0, 1]	2
1	[0, 1]	2
2	[2]	1

Najmanja vrijednost $p[i]$ po svim sobama je 1, a to se postiže kada $i = 2$. Stoga bi ovaj postupak trebao vratiti [0, 0, 1].

Ograničenja

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

Podzadaci

1. (9 bodova) $c[j] = 0$ za sve $0 \leq j \leq m - 1$ i $n, m \leq 200$

2. (11 bodova) $n, m \leq 200$
3. (17 bodova) $n, m \leq 2000$
4. (30 bodova) $c[j] \leq 29$ (za sve $0 \leq j \leq m - 1$) i $r[i] \leq 29$ (za sve $0 \leq i \leq n - 1$)
5. (33 boda) Nema dodatnih ograničenja.

Sample Grader

Sample Grader čita ulaz u sljedećem formatu:

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

Sample Grader ispisuje vraćenu vrijednost procedure `find_reachable` u sljedećem formatu:

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