

Açarlar

Memar Timati yeni bir qaçış oyunu hazırladı. Bu oyunda 0 -dan $n - 1$ -ə nömrələnmiş n sayda otaq var. Başlanğıcda hər bir otaqda tam olaraq bir açar var. Hər bir açarın 0 ilə $n - 1$ arasında (hər ikisi daxil) bir tam ədəd ilə işarələnmiş bir növü var. i -ci ($0 \leq i \leq n - 1$) otaqdakı açarın növü $r[i]$ -dir. Qeyd edək ki, müxtəlif otaqlarda eyni növdə açarlar ola bilər, yəni ki, $r[i]$ dəyərləri mütləq deyil ki hamısı fərqli olsun.

Oyunda həmçinin 0 ilə $m - 1$ arasında nömrələnmiş m sayda **qoşa istiqamətli** birləşdirici var. j ($0 \leq j \leq m - 1$) nömrəli birləşdirici $u[j]$ və $v[j]$ nömrəli bir cüt müxtəlif otağı bir-birinə bağlayır. Bir cüt otaq birdən çox birləşdirici ilə bir-birinə bağlanmış ola bilər.

Oyun tək oyunçu tərəfindən oynanılır. Oyunçu açarlar toplayır və birləşdiricilərdən istifadə edərək otaqlar arasında hərəkət edir. Oyunçu j nömrəli birləşdirici ilə **hərəkət edir** dedikdə, onun $u[j]$ nömrəli otaqdan $v[j]$ nömrəli otağa və ya tərsinə getdiyi nəzərdə tutulur. Oyunçu j nömrəli birləşdiricidə o zaman hərəkət edə bilər ki, o $c[j]$ növlü açarı öncədən toplamış olsun.

Oyunda hər hansı bir anda oyunçu hər hansı x nömrəli otaqda olarsa aşağıdakı iki hərəkəti edə bilər:

- x nömrəli otaqdakı $r[x]$ növlü açarı toplamaq (əgər hələ də toplamayıbsa)
- j nömrəli birləşdiricidə hərəkət etmək, əgər $u[j] = x$ və ya $v[j] = x$ olarsa və oyunçu artıq $c[j]$ növlü açarı öncədən toplamayıbsa. Qeyd edək ki, oyunçu topladığı açarları **heç vaxt** atmır.

Oyunçu heç bir açar olmadan hər hansı s nömrəli otaqda oyuna **başlayır**. t nömrəli otaq s nömrəli otaqdan o zaman çatıla bilən sayılır ki, oyunçu s nömrəli otaqda başlayıb yuxarıda təsvir edilən addımları bir neçə dəfə etməklə t nömrəli otağa çata bilsin.

Hər bir i ($0 \leq i \leq n - 1$) nömrəli otaq üçün, ondan gedilə bilən otaqların sayına $p[i]$ deyək. Timati $p[i]$ dəyəri minimum olan i ($0 \leq i \leq n - 1$) nömrəli otaqları bilmək istəyir.

İmplementasiya detalları

You are to implement the following procedure:

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

- r : an array of length n . For each i ($0 \leq i \leq n - 1$), the key in room i is of type $r[i]$.
- u, v : two arrays of length m . For each j ($0 \leq j \leq m - 1$), connector j connects rooms $u[j]$ and $v[j]$.
- c : an array of length m . For each j ($0 \leq j \leq m - 1$), the type of key needed to traverse connector j is $c[j]$.

- This procedure should return an array s of length n . For each $0 \leq i \leq n - 1$, the value of $s[i]$ should be 1 if for every j such that $0 \leq j \leq n - 1$, $p[i] \leq p[j]$. Otherwise, the value of $s[i]$ should be 0.

Nümunələr

Nümunə 1

Consider the following call:

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

If the player starts the game in room 0, they can perform the following sequence of actions:

Current room	Action
0	Collect key of type 0
0	Traverse connector 0 to room 1
1	Collect key of type 1
1	Traverse connector 2 to room 2
2	Traverse connector 2 to room 1
1	Traverse connector 3 to room 3

Hence room 3 is reachable from room 0. Similarly, we can construct sequences showing that all rooms are reachable from room 0, which implies $p[0] = 4$. The table below shows the reachable rooms for all starting rooms:

Starting room i	Reachable rooms	$p[i]$
0	[0, 1, 2, 3]	4
1	[1, 2]	2
2	[1, 2]	2
3	[1, 2, 3]	3

The smallest value of $p[i]$ across all rooms is 2, and this is attained for $i = 1$ or $i = 2$. Therefore, this procedure should return [0, 1, 1, 0].

Nümunə 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])
```

The table below shows the reachable rooms:

Starting room i	Reachable rooms	$p[i]$
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

The smallest value of $p[i]$ across all rooms is 2, and this is attained for $i \in \{1, 2, 4, 6\}$. Therefore, this procedure should return [0, 1, 1, 0, 1, 0, 1].

Nümunə 3

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

The table below shows the reachable rooms:

Starting room i	Reachable rooms	$p[i]$
0	[0, 1]	2
1	[0, 1]	2
2	[2]	1

The smallest value of $p[i]$ across all rooms is 1, and this is attained when $i = 2$. Therefore, this procedure should return [0, 0, 1].

Məhdudiyyətlər

- $2 \leq n \leq 300\,000$
- $1 \leq m \leq 300\,000$
- $0 \leq r[i] \leq n - 1$ bütün $0 \leq i \leq n - 1$ üçün
- $0 \leq u[j], v[j] \leq n - 1$ və $u[j] \neq v[j]$ bütün $0 \leq j \leq m - 1$ üçün

- $0 \leq c[j] \leq n - 1$ bütün $0 \leq j \leq m - 1$ üçün

Alt tapşırıqlar

1. (9 bal) $c[j] = 0$ bütün $0 \leq j \leq m - 1$ üçün və $n, m \leq 200$
2. (11 bal) $n, m \leq 200$
3. (17 bal) $n, m \leq 2000$
4. (30 bal) $c[j] \leq 29$ (bütün $0 \leq j \leq m - 1$ üçün) və $r[i] \leq 29$ (bütün $0 \leq i \leq n - 1$ üçün)
5. (33 bal) Əlavə məhdudiyyət yoxdur.

Nümunə grader (qiymətləndirici)

Nümunə grader inputu aşağıdakı formatda oxuyur:

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

Nümunə grader `find_reachable` prosedurundan qayıdan dəyəri aşağıdakı formatda çap edir:

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