

Клучеви

Архитектот Јованов разбрал дека тимот за ИОИ планира да оди на Escape Room, па одлучил да дизајнира нова "Escape" игра. Во оваа игра, дадени се n соби означени од 0 до $n - 1$. Иницијално, секоја соба содржи точно еден клуч. Секој клуч има тип, кој што е цел број помеѓу 0 и $n - 1$, вклучително. Типот на клучот во соба i ($0 \leq i \leq n - 1$) е $r[i]$. Забележете дека повеќе соби можат да содржат клуч од истиот тип, т.е., вредностите $r[i]$ не мора да бидат различни.

Исто така има m **двонасочни** конектори во играта, означени од 0 до $m - 1$. Конекторот j ($0 \leq j \leq m - 1$) ги поврзува парот различни соби $u[j]$ и $v[j]$. Еден пар од соби може да биде поврзан со повеќе конектори.

Играта се игра од еден играч кој што ги собира клучевите и се движи меѓу собите минувајќи низ конекторите. Ќе кажеме дека играчот **минува низ** конекторот j кога го користи овој конектор за да се движи од собата $u[j]$ во собата $v[j]$, или обратно од собата $v[j]$ во собата $u[j]$. Играчот може да минува низ конекторот j ако претходно има соберено клуч од тип $c[j]$.

Во било кој момент за време на играта, играчот е во некоја соба x и може да изврши два вида на акции:

- да го собере клучот кој се наоѓа во соба x , чиј што тип е $r[x]$ (освен ако веќе не има соберено клуч од тој тип),
- да помине низ конекторот j , каде или $u[j] = x$ или $v[j] = x$, ако играчот има соберено клуч од тип $c[j]$ пред ова движење. Забележете дека играчот **никогаш** не ги отфрла клучевите кои што ги собира.

Играчот ја **започнува** играта во некоја соба s без било кој клуч. Соба t е **достижна** од соба s , ако играч кој што ја започнал играта во соба s може да изврши некоја секвенца од акциите дадени погоре, и да стаса во собата t .

За секоја соба i ($0 \leq i \leq n - 1$), го означуваме бројот на соби достижни од собата i со $p[i]$. Јованов би сакал да знае кое е множеството на индекси i кои што имаат минимална вредност $p[i]$ од индексите $0 \leq i \leq n - 1$.

Детали за имплементација

Вие треба да ја имплементирате следната процедура.

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

- r : е низа со должина n . За секој i ($0 \leq i \leq n - 1$), клучот во соба i е од тип $r[i]$.

- u, v : две низи со должина m . За секој j ($0 \leq j \leq m - 1$), конекторот j ги поврзува собите $u[j]$ и $v[j]$.
- c : низа со должина m . За секој j ($0 \leq j \leq m - 1$), типот на клучот потребен за да се минува низ конекторот j е $c[j]$.
- Оваа процедура треба да врати низа a со должина n . За секој $0 \leq i \leq n - 1$, вредноста на $a[i]$ треба да биде 1 ако за секој j т.д. $0 \leq j \leq n - 1$, важи $p[i] \leq p[j]$. Инаку, вредноста на $a[i]$ треба да биде 0.

Примери

Пример 1

Да го разгледаме следниот повик:

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

Ако играчот ја започне играта во соба 0, тие можат да ја извршат следната секвенца на акции:

Моментална соба	Акција
0	Собери го клучот од тип 0
0	Помини низ конекторот 0 во соба 1
1	Собери го клучот од тип 1
1	Помини низ конекторот 2 во соба 2
2	Помини низ конекторот 2 во соба 1
1	Помини низ конекторот 3 во соба 3

Оттука собата 3 е достижна од собата 0. Слично, можеме да конструираме секвенци за да покажеме дека сите соби се достижни од соба 0, што подразбира $p[0] = 4$. Табелата подолу ги покажува достижните соби од сите соби во кои можеме да почнеме.

Почетна соба i	Достижни соби	$p[i]$
0	[0, 1, 2, 3]	4
1	[1, 2]	2
2	[1, 2]	2
3	[1, 2, 3]	3

Најмалата вредност за $p[i]$ од сите соби е 2, и се добива за $i = 1$ или $i = 2$. Затоа, оваа процедура треба да врати [0, 1, 1, 0].

Пример 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])
```

Табелата подолу ги покажува достижните соби:

Почетна соба i	Достижни соби	$p[i]$
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

Најмалата вредност за $p[i]$ од сите соби е 2, и се добива за $i \in \{1, 2, 4, 6\}$. Затоа, оваа процедура треба да врати [0, 1, 1, 0, 1, 0, 1].

Пример 3

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

Табелата подолу ги покажува достижните соби:

Почетна соба i	Достижни соби	$p[i]$
0	[0, 1]	2
1	[0, 1]	2
2	[2]	1

Најмалата вредност за $p[i]$ од сите соби е 1, и се добива кога $i = 2$. Затоа, оваа процедура треба да врати [0, 0, 1].

Ограничувања

- $2 \leq n \leq 300\,000$
- $1 \leq m \leq 300\,000$

- $0 \leq r[i] \leq n - 1$ за сите $0 \leq i \leq n - 1$
- $0 \leq u[j], v[j] \leq n - 1$ и $u[j] \neq v[j]$ за сите $0 \leq j \leq m - 1$
- $0 \leq c[j] \leq n - 1$ за сите $0 \leq j \leq m - 1$

Подзадачи

1. (9 поени) $c[j] = 0$ за сите $0 \leq j \leq m - 1$ и $n, m \leq 200$
2. (11 поени) $n, m \leq 200$
3. (17 поени) $n, m \leq 2000$
4. (30 поени) $c[j] \leq 29$ (за сите $0 \leq j \leq m - 1$) и $r[i] \leq 29$ (за сите $0 \leq i \leq n - 1$)
5. (33 поени) Нема дополнителни ограничувања.

Оценувач

Дадениот оценувач го чита влезот во следниот формат:

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

Дадениот оценувач ја печати вратената вредност од `find_reachable` во следниот формат:

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