

Ключи

Архитектор Тимати разработал новую игру в стиле квеста. В этой игре есть n комнат, пронумерованных от 0 до $n - 1$. Изначально в каждой комнате расположен ровно один ключ. Каждый из ключей имеет один из типов, пронумерованных от 0 до $n - 1$. Тип ключа в комнате i ($0 \leq i \leq n - 1$) равен $r[i]$. Обратите внимание, что несколько комнат могут содержать ключи одного и того же типа, то есть, значения $r[i]$ не обязательно являются различными.

В игре также есть m **двусторонних** переходов, пронумерованных от 0 до $m - 1$. Переход j ($0 \leq j \leq m - 1$) соединяет пару различных комнат $u[j]$ и $v[j]$. Одна и та же пара комнат может быть соединена несколькими переходами.

В игре участвует один человек, который собирает ключи и перемещается между комнатами с помощью переходов. Будем говорить, что игрок **проходит** переход j , если игрок перемещается по этому переходу от $u[j]$ до $v[j]$ или в обратном направлении. Игрок может пройти через переход j только в том случае, если у игрока уже есть ключ типа $c[j]$.

В любой момент игры игрок находится в некоторой комнате x и может выполнить одно из следующих действий:

- собрать ключ в комнате x , тип которого равен $r[x]$ (если игрок этого еще не сделал),
- пройти через переход j , где или $u[j] = x$, или $v[j] = x$, если у игрока уже есть ключ типа $c[j]$. Обратите внимание, что игрок **никогда** не избавляется от ключей, которые уже были найдены.

Игрок **начинает** игру в комнате с номером s без единого ключа. Комната t называется **достижимой** из комнаты s в том случае, если игрок, начав в комнате s , может добраться до комнаты t , выполняя действия, описанные выше.

Для каждой комнаты i ($0 \leq i \leq n - 1$) обозначим как $p[i]$ общее число комнат, достижимых из i . Тимати хочет найти множество индексов i , на которых достигается минимальное значение $p[i]$ среди $0 \leq i \leq n - 1$.

Детали реализации

Вам необходимо реализовать следующую функцию:

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

- r : массив длины n . Для каждого i ($0 \leq i \leq n - 1$) ключ в комнате i имеет тип $r[i]$.
- u, v : два массива длины m . Для каждого j ($0 \leq j \leq m - 1$) переход j соединяет комнаты $u[j]$ и $v[j]$.

- c : массив длины m . Для каждого j ($0 \leq j \leq m - 1$) указан тип ключа $c[j]$, необходимый для того, чтобы пройти переход j .
- Данная функция должна возвращать массив a длины n . Для каждого $0 \leq i \leq n - 1$ значение $a[i]$ должно быть равно 1 в том случае, если для всех j , для которых $0 \leq j \leq n - 1$, выполнено $p[i] \leq p[j]$. В противном случае значение $a[i]$ должно быть равно 0.

Примеры

Пример 1

Рассмотрим следующий вызов функции

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

Если игрок начинает в комнате 0, то игрок может выполнить следующую последовательность действий:

Текущая комната	Действие
0	Собрать ключ типа 0
0	Перейти по переходу 0 в комнату 1
1	Собрать ключ типа 1
1	Перейти по переходу 2 в комнату 2
2	Перейти по переходу 2 в комнату 1
1	Перейти по переходу 3 в комнату 3

Таким образом, комната 3 достижима из комнаты 0. Аналогичным образом мы можем построить последовательности действий для каждой из комнат, таким образом, все они являются достижимыми из комнаты 0, что означает, что $p[0] = 4$. В таблице ниже приведены достижимые комнаты для каждого варианта стартовой комнаты:

Стартовая комната i	Достижимые комнаты	$p[i]$
0	[0, 1, 2, 3]	4
1	[1, 2]	2
2	[1, 2]	2
3	[1, 2, 3]	3

Наименьшее значение $p[i]$ среди всех комнат равно 2, и оно достигается для $i = 1$ и $i = 2$. Таким образом, функция должна вернуть [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])
```

В таблице ниже приведены достижимые комнаты для каждого варианта стартовой комнаты:

Стартовая комната i	Достижимые комнаты	$p[i]$
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

Наименьшее значение $p[i]$ среди всех комнат равно 2, и это значение достигается для $i \in \{1, 2, 4, 6\}$. Таким образом, функция должна вернуть [0, 1, 1, 0, 1, 0, 1].

Пример 3

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

В таблице ниже приведены достижимые комнаты для каждого варианта стартовой комнаты:

Стартовая комната i	Достижимые комнаты	$p[i]$
0	[0, 1]	2
1	[0, 1]	2
2	[2]	1

Наименьшее значение $p[i]$ среди всех комнат равно 1, и это значение достигается при $i = 2$. Таким образом, функция должна вернуть [0, 0, 1].

Ограничения

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

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

Подзадачи

1. (9 баллов) $c[j] = 0$ для всех $0 \leq j \leq m - 1$ и $n, m \leq 200$
2. (11 баллов) $n, m \leq 200$
3. (17 баллов) $n, m \leq 2000$
4. (30 баллов) $c[j] \leq 29$ (для всех $0 \leq j \leq m - 1$) и $r[i] \leq 29$ (для всех $0 \leq i \leq n - 1$)
5. (33 балла) Без дополнительных ограничений.

Пример проверяющего модуля

Проверяющий модуль считывает данные в следующем формате:

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

Проверяющий модуль выводит результат вызова функции `find_reachable` в следующем формате:

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