Keys

В новата игра за бягство има n стаи, номерирани от 0 до $n-1$. В началото всяка стая съдържа точно един ключ. Всеки ключ има тип, който е цяло число между 0 и $n-1$, включително. Типът на ключа в стая i ($0 \leq i \leq n-1$) е $r[i]$. Възможно е няколко стаи да съдържат ключове от един и същ тип, т.е. стойностите $r[i]$ не са задължително различни.

В играта има m **двустранни** връзки, номерирани от 0 до $m-1$. Връзката j ($0 \leq j \leq m-1$) свързва двойка различни стаи $u[j]$ и $v[j]$. Две стаи могат да бъдат свързани и с повече от една връзка.

Играта се играе от един играч, който събира ключовете и се придвижва от стая в стая, минавайки по връзките.

Казваме, че играчът **преминава** по връзката j, когато той използва тази връзка, за да отиде от стая $u[j]$ в стая $v[j]$, или обратното. Играчът може да премине по връзка j само ако преди това той е взел ключ от тип $c[j]$.

Във всеки момент от играта, играчът е в някоя стая x и може да извърши два вида действия:

- да вземе ключът от стая x, чийто тип е $r[x]$ (ако вече не е бил взет),
- да премине по връзка j, при което или $u[j] = x$ или $v[j] = x$, като това може да стане, ако играчът вече е взел ключ от тип $c[j]$. Играчът **никога** не изхвърля ключ, който е взел.

Играчът **започва** играта в някоя стая s, като няма в себе си нито един ключ.

Стая t е **достижима** от стая s, ако играчът, който е започнал играта от стая s може да изпълни последователност от действия, описани по-горе, така че да отиде в стая t.

За всяка стая i ($0 \leq i \leq n-1$) означаваме с $p[i]$ броя на стаите, които са достижими от стая i. Искаме да намерим тези индекси i ($0 \leq i \leq n-1$), за които се получава минимална стойност на $p[i]$.

Детайли по реализацията

Вие трябва да реализирате следната функция:

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

- r: масив с дължина n. За всяко i ($0 \leq i \leq n-1$), ключът в стая i е от тип $r[i]$.
- u, v: два масива с дължина m. За всяко j ($0 \leq j \leq m-1$), връзката j свързва стаи $u[j]$ и $v[j]$.

- c: масив с дължина m. За всяко j ($0 \leq j \leq m-1$), типът на ключа, необходим да се премине по връзка j е $c[j]$.
- Фукцията трябва да върне масив s с дължина n. За всяко $0 \leq i \leq n-1$, стойността на $a[i]$ трябва да е 1, ако за всяко $0 \leq j \leq n-1$, $p[i] \leq p[j]$. В противен случай стойността на $a[i]$ трябва да е 0.

Примери

Пример 1

Разглеждаме следното извикване:

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

Ако играчът започне играта от стая 0, той може да изпълни следната последователност от действия:

Текуща стая	Действие
0	Взема ключ от тип 0
0	Преминава по връзка 0 към стая 1
1	Взема ключ от тип 1
1	Преминава по връзка 2 към стая 2
2	Преминава по връзка 2 към стая 1
1	Преминава по връзка 3 към стая 3

Следователно, стая 3 е достижима от стая 0. Аналогично, може да конструираме последователности от действия, с които да покажем, че всички стаи са достижими от стая 0, от което следва, че $p[0] = 4$. Таблицата по-долу показва достижимите стаи, при тръгване от всяка възможна начална стая:

Начална стая i	Достижими стаи	$p[i]$
0	$[0, 1, 2, 3]$	4
1	$[1, 2]$	2
2	$[1, 2]$	2
3	$[1, 2, 3]$	3

Най-малката стойност на $p[i]$ в таблицата е 2 и тази стойност се достига за $i = 1$ и $i = 2$. Следователно, функцията find_reachable трябва да върне $[0, 1, 1, 0]$.

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])
```

Таблицата по-долу показва достижимите стаи:

Начална стая i	**Достижими стаи**	$p[i]$
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

Най-малката стойност на $p[i]$ е 2 и се достига за $i \in \{1, 2, 4, 6\}$. Следователно, функцията find_reachable трябва да върне $[0, 1, 1, 0, 1, 0, 1]$.

Пример 3

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

Таблицата по-долу показва достижимите стаи:

Начална стая i	**Достижими стаи**	$p[i]$
0	$[0, 1]$	2
1	$[0, 1]$	2
2	$[2]$	1

Най-малката стойност на $p[i]$ е 1 и се достига за $i = 2$. Следователно, функцията find_reachable трябва да върне $[0, 0, 1]$.

Ограничения

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

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

Подзадачи

1. (9 точки) $c[j] = 0$ за $0 \leq j \leq m-1$ и $n, m \leq 200$
2. (11 точки) $n, m \leq 200$
3. (17 точки) $n, m \leq 2000$
4. (30 точки) $c[j] \leq 29$ (за $0 \leq j \leq m-1$) и $r[i] \leq 29$ (за $0 \leq i \leq n-1$)
5. (33 точки) няма допълнителни ограничения.

Примерен грейдер

Примерният грейдер чете входа в следния формат:

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

Примерният грейдер отпечатва стойността, която връща `find_reachable` в следния формат:

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