

Sphinx's Riddle

У Сфинкса есть загадка для вас. Вам дан граф с N вершинами. Вершины пронумерованы от 0 до $N - 1$. Граф содержит M ребер, пронумерованных от 0 до $M - 1$. Каждое ребро соединяет пару различных вершин и является неориентированным. А именно, для каждого j от 0 до $M - 1$, включительно, ребро j соединяет вершины $X[j]$ и $Y[j]$. Каждая пара вершин соединена не более чем одним ребром. Две вершины являются **смежными**, если они соединены ребром.

Последовательность вершин v_0, v_1, \dots, v_k (для $k \geq 0$) называется **путем**, если любые две последовательные вершины v_l и v_{l+1} (для всех l , для которых $0 \leq l < k$) являются смежными. В этом случае мы говорим, что путь v_0, v_1, \dots, v_k **соединяет** вершины v_0 и v_k . В данном вам графе любая пара вершин соединена каким-либо путем.

Есть $N + 1$ цветов, пронумерованных от 0 до N . Цвет N особенный и называется **цвет Сфинкса**. Каждая вершина покрашена в некоторый цвет. А именно, вершина i ($0 \leq i < N$) покрашена в цвет $C[i]$. Несколько вершин могут быть покрашены в один и тот же цвет, также могут быть цвета, в которые не покрашены никакие вершины. Никакая вершина не покрашена в цвет Сфинкса, то есть, $0 \leq C[i] < N$ ($0 \leq i < N$).

Путь v_0, v_1, \dots, v_k (для $k \geq 0$) называется **монохромным**, если все его вершины покрашены в один и тот же цвет, то есть $C[v_l] = C[v_{l+1}]$ (для всех l , таких что $0 \leq l < k$). Будем говорить, что вершины p и q ($0 \leq p < N$, $0 \leq q < N$) находятся в одной **монохромной компоненте**, если и только если они соединены монохромным путем.

Вам даны вершины и ребра графа, но вы не знаете, в какой цвет раскрашена какая вершина. Вы хотите найти цвета вершин, используя **эксперименты по перекраске**.

В эксперименте по перекраске вы можете временно перекрасить произвольное множество вершин. А именно, для выполнения эксперимента по перекраске вы сначала выбираете массив E длины N , где для каждого i ($0 \leq i < N$) $E[i]$ находится в диапазоне -1 и N , **включительно**. После этого цвет вершины i становится равен $S[i]$, где значение $S[i]$ равно:

- $C[i]$, то есть исходному цвету вершины i , если $E[i] = -1$, либо
- $E[i]$, в противном случае.

Обратите внимание, что в эксперименте по перекраске вы можете использовать цвет Сфинкса.

После перекраски Сфинкс сообщает вам количество монохромных компонент графа, получившегося после перекраски вершины i в цвет $S[i]$ (для всех i , таких что $0 \leq i < N$), и **возвращает вершинам их исходные цвета**.

Ваша задача — определить исходные цвета всех вершин графа, выполнив не более 2 750 экспериментов по перекраске. Вы можете также получить частичный балл, если вы корректно определите для каждой пары смежных вершин, верно ли, что они имеют одинаковый цвет.

Implementation Details

Вам следует реализовать следующую функцию

```
std::vector<int> find_colours(int N,  
    std::vector<int> X, std::vector<int> Y)
```

- N : количество вершин графа.
- X, Y : массивы длины M , задающие ребра графа.
- Функция должна вернуть массив G длины N , описывающий исходные цвета вершин графа.
- Эта функция будет вызвана ровно один раз для каждого теста.

Описанная выше функция может делать вызовы следующей функции:

```
int perform_experiment(std::vector<int> E)
```

- E : массив длины N , задающий описание перекраски вершин.
- Функция возвращает количество монохромных компонент в графе в эксперименте по перекраске с использованием массива E .
- Эту функцию разрешается вызывать не более 2 750 раз.

Грейдер в этой задаче **не адаптивный**, иначе говоря, цвета всех вершин зафиксированы до выполнения вызова `find_colours`.

Constraints

- $2 \leq N \leq 250$
- $N - 1 \leq M \leq \frac{N \cdot (N - 1)}{2}$
- $0 \leq X[j] < Y[j] < N$ для всех j , таких что $0 \leq j < M$.
- $X[j] \neq X[k]$ или $Y[j] \neq Y[k]$ для всех j и k , таких что $0 \leq j < k < M$.
- Каждая пара вершин соединена каким-либо путем.
- $0 \leq C[i] < N$ для всех i , таких что $0 \leq i < N$.

Subtasks

Подзадача	Балл	Дополнительные ограничения
1	3	$N = 2$
2	7	$N \leq 50$
3	33	Граф является путем: $M = N - 1$, и вершины j и $j + 1$ являются смежными ($0 \leq j < M$).
4	21	Граф является полным: $M = \frac{N \cdot (N - 1)}{2}$, и любые две вершины являются смежными.
5	36	Без дополнительных ограничений.

В каждой подзадаче вы можете получить частичный балл, если ваша программа определит корректно для каждой пары смежных вершин, раскрашены ли они в один и тот же цвет.

А именно, вы получите полный балл за подзадачу, если во всех тестах этой подзадачи массив G , который возвращает `find_colours`, в точности совпадает с массивом C (то есть $G[i] = C[i]$ для всех i , таких что $0 \leq i < N$). Иначе вы получите 50% баллов за подзадачу, если во всех тестах этой подзадачи выполнены следующие условия:

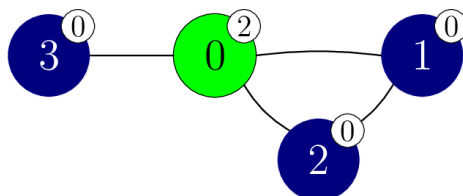
- $0 \leq G[i] < N$ для всех i , таких что $0 \leq i < N$;
- Для всех j , таких что $0 \leq j < M$:
 - $G[X[j]] = G[Y[j]]$, если и только если $C[X[j]] = C[Y[j]]$.

Example

Рассмотрим следующий вызов.

```
find_colours(4, [0, 1, 0, 0], [1, 2, 2, 3])
```

Для этого примера положим, что (не известные вам) цвета вершин заданы как $C = [2, 0, 0, 0]$. Этот сценарий показан на следующем рисунке. Цвета дополнительно обозначены числами в белых метках возле вершин



Функция может вызвать `perform_experiment` следующим образом.

```
perform_experiment([-1, -1, -1, -1])
```

В этом вызове ни одна вершина не перекрашивается, таким образом все вершины сохраняют свои оригинальные цвета.

Рассмотрим вершины 1 и 2. У них обеих цвет 0 и путь 1,2 монохромный. Вершины 1 и 2 находятся в одной монохромной компоненте.

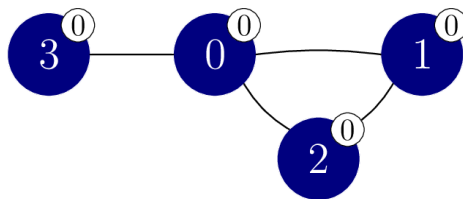
Рассмотрим вершины 1 и 3. Несмотря на то, что обе они одного цвета 0, они находятся в разных монохромных компонентах, поскольку их не соединяет монохромный путь.

В итоге всего есть 3 монохромных компоненты с вершинами $\{0\}$, $\{1,2\}$ и $\{3\}$. Таким образом, этот вызов возвращает 3.

Теперь функция может вызвать `perform_experiment` следующим образом.

```
perform_experiment([0, -1, -1, -1])
```

В этом вызове только одна вершина 0 перекрашена в цвет 0, результат перекраски показан на следующем рисунке.

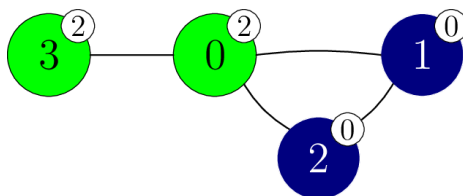


Этот вызов возвращает 1, поскольку все вершины принадлежат одной монохромной компоненте. Можно сделать вывод, что вершины 1, 2 и 3 имеют цвет 0.

Затем функция может вызвать `perform_experiment` следующим образом.

```
perform_experiment([-1, -1, -1, 2])
```

В этом вызове вершина 3 перекрашена в цвет 2, результат перекраски показан на следующем рисунке.



Этот вызов возвращает 2, поскольку есть 2 монохромные компоненты, с вершинами $\{0,3\}$ и $\{1,2\}$ соответственно. Можно сделать вывод, что вершина 0 имеет цвет 2.

Функция `find_colours` затем возвращает массив $[2, 0, 0, 0]$. Поскольку $C = [2, 0, 0, 0]$, получен полный балл.

Также есть несколько возвращаемых значений, для которых будут получены 50% от полного балла, например, $[1, 2, 2, 2]$ или $[1, 2, 2, 3]$.

Sample Grader

Input format:

```
N M
C[0] C[1] ... C[N-1]
X[0] Y[0]
X[1] Y[1]
...
X[M-1] Y[M-1]
```

Output format:

```
L Q
G[0] G[1] ... G[L-1]
```

Здесь L это длина массива G , возвращённая функцией `find_colours`, а Q количество вызовов функции `perform_experiment`.