



Beech Tree

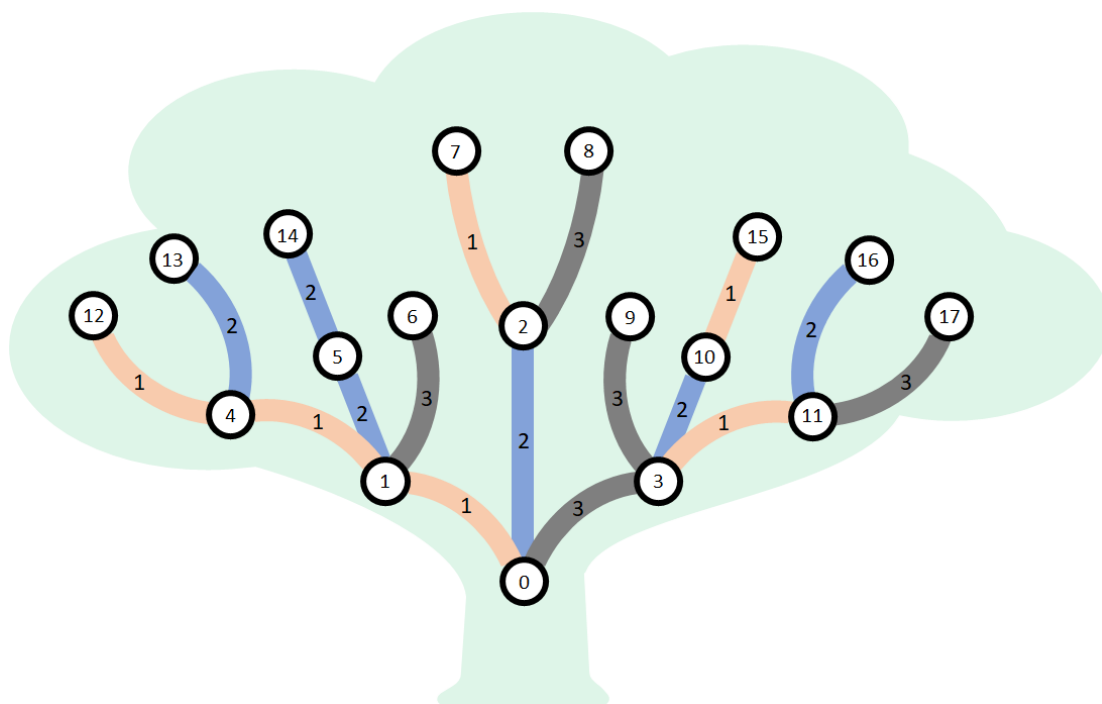
Ветьемский лес — знаменитый лесной массив с множеством разноцветных деревьев. Одно из старейших и самых высоких буковых деревьев называется Ос Везер.

Дерево Ос Везер может быть описано множествами из N **вершин** и $N - 1$ **ребер**. Вершины пронумерованы от 0 до $N - 1$ и ребра пронумерованы от 1 до $N - 1$. Каждое ребро соединяет две различных вершины дерева. Ребро i ($1 \leq i < N$) соединяет вершину i с вершиной $P[i]$, где $0 \leq P[i] < i$. Вершина $P[i]$ называется **родителем** вершины i и вершина i называется **сыном** вершины $P[i]$.

Каждое ребро имеет цвет. Всего есть M различных цветов, они пронумерованы от 1 до M . Цвет ребра i это $C[i]$. Различные ребра могут иметь одинаковый цвет.

Обратите внимание, что в определении выше $i = 0$ не соответствует ребру дерева. Для удобства будем считать что $P[0] = -1$ и $C[0] = 0$.

Например, предположим, что Ос Везер содержит $N = 18$ вершин и есть $M = 3$ возможных цвета ребер. Ребра описываются последовательностью $P = [-1, 0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 3, 4, 4, 5, 10, 11, 11]$, их цвета последовательностью $C = [0, 1, 2, 3, 1, 2, 3, 1, 3, 3, 2, 1, 1, 2, 2, 1, 2, 3]$. Ниже приведено изображение дерева:



Арпад это талантливый лесник, который изучает отдельные части дерева, называемые **поддеревьями**. Для всех r , таких что $0 \leq r < N$, поддерево вершины r это множество вершин $T(r)$, обладающее следующими свойствами:

- Вершина r лежит в $T(r)$.
- Если вершина x лежит в $T(r)$, все сыновья x тоже лежат в $T(r)$.
- Никакие другие вершины не лежат в $T(r)$.

Размер множества $T(r)$ обозначим как $|T(r)|$.

Арпад недавно открыл сложное, но интересное свойство поддеревьев. Открытие Арпада потребовало от него много работы с ручкой и бумагой, и он подозревает, что вам, возможно, придется сделать то же самое, чтобы понять его. Он покажет вам несколько примеров, которые вы можете детально проанализировать.

Пусть мы зафиксировали r и перестановку $v_0, v_1, \dots, v_{|T(r)|-1}$ вершин $T(r)$.

Для всех i таких что $1 \leq i < |T(r)|$, обозначим за $f(i)$ количество раз, которое цвет $C[v_i]$ встречается в следующей последовательности из $i - 1$ цветов: $C[v_1], C[v_2], \dots, C[v_{i-1}]$.

(Обратите внимание, что $f(1)$ всегда равно 0, потому что последовательность цветов в определении пуста.)

Перестановка $v_0, v_1, \dots, v_{|T(r)|-1}$ **красивая** тогда и только тогда, когда следующие свойства выполнены:

- $v_0 = r$.
- Для всех i таких что $1 \leq i < |T(r)|$, родитель вершины v_i это вершина $v_{f(i)}$.

Для всех r таких что $0 \leq r < N$, поддерево $T(r)$ **красивое** тогда и только тогда, когда существует красивая перестановка вершин поддерева $T(r)$. Обратите внимание, что по определению каждое поддерево, которое состоит из единственной вершины красивое.

Рассмотрим пример дерева выше. Поддеревья $T(0)$ и $T(3)$ этого дерева некрасивые. Поддерево $T(14)$ красивое, потому что оно содержит единственную вершину. Ниже мы показываем, что поддерево $T(1)$ также красивое.

Рассмотрим последовательность различных целых чисел $[v_0, v_1, v_2, v_3, v_4, v_5, v_6] = [1, 4, 5, 12, 13, 6, 14]$. Эта последовательность это перестановка вершин поддерева $T(1)$. Эта перестановка показана на изображении снизу. Номер каждой вершины в этой перестановке это число, записанное на табличке рядом с вершиной.


```
int[] beechtree(int N, int M, int[] P, int[] C)
```

- N : количество вершин в дереве.
- M : количество возможных цветов ребер.
- P, C : массивы длины N , описывающие ребра дерева.
- Эта функция должна вернуть массив b длины N . Для всех r , таких что $0 \leq r < N$, $b[r]$ должно быть равно 1, если $T(r)$ красивое поддерево и 0, иначе.
- Эта функция вызывается ровно один раз для каждого теста.

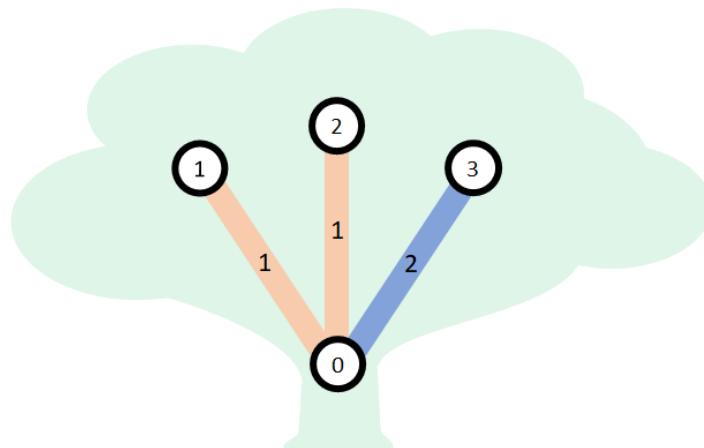
Examples

Example 1

Рассмотрим следующий вызов:

```
beechtree(4, 2, [-1, 0, 0, 0], [0, 1, 1, 2])
```

Это дерево изображено ниже:



Поддеревья $T(1)$, $T(2)$ и $T(3)$ состоят из единственной вершины и поэтому являются красивыми. Поддерево $T(0)$ некрасивое. Таким образом, функция должна вернуть $[0, 1, 1, 1]$.

Example 2

Рассмотрим следующий вызов:

```
beechtree(18, 3,  
    [-1, 0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 3, 4, 4, 5, 10, 11, 11],  
    [0, 1, 2, 3, 1, 2, 3, 1, 3, 3, 2, 1, 1, 2, 2, 1, 2, 3])
```

Это дерево изображено выше в условии задачи.

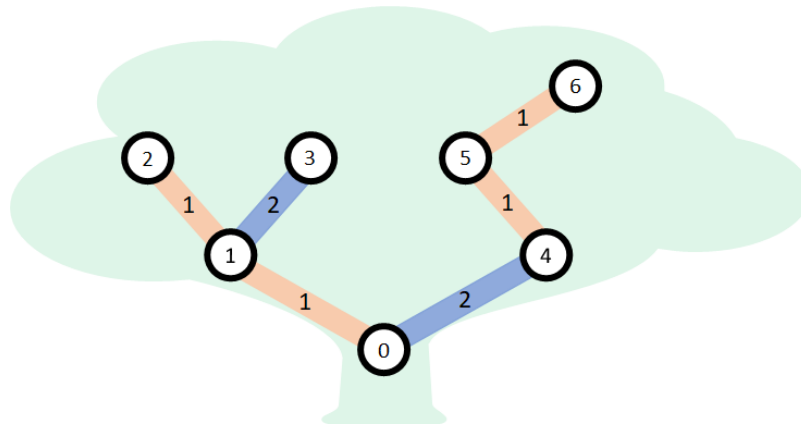
Функция должна вернуть $[0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$.

Example 3

Рассмотрим следующий вызов:

```
beechtree(7, 2, [-1, 0, 1, 1, 0, 4, 5], [0, 1, 1, 2, 2, 1, 1])
```

Это дерево изображено ниже:



$T(0)$ это единственное некрасивое поддереву. Функция должна вернуть $[0, 1, 1, 1, 1, 1, 1]$.

Constraints

- $3 \leq N \leq 200\,000$
- $2 \leq M \leq 200\,000$
- $0 \leq P[i] < i$ (для всех i таких что $1 \leq i < N$)
- $1 \leq C[i] \leq M$ (для всех i таких что $1 \leq i < N$)
- $P[0] = -1$ и $C[0] = 0$

Subtasks

1. (9 баллов) $N \leq 8$ и $M \leq 500$
2. (5 баллов) Ребро i соединяет вершину i с вершиной $i - 1$. То есть для всех i таких что $1 \leq i < N$, $P[i] = i - 1$.
3. (9 баллов) Каждая вершина, кроме вершины 0 либо соединена с вершиной 0 или соединена с вершиной, которая соединена с вершиной 0. То есть для всех i таких что $1 \leq i < N$, или $P[i] = 0$ или $P[P[i]] = 0$.
4. (8 баллов) Для всех c таких что $1 \leq c \leq M$ есть не более двух ребер цвета c .
5. (14 баллов) $N \leq 200$ и $M \leq 500$
6. (14 баллов) $N \leq 2\,000$ и $M = 2$
7. (12 баллов) $N \leq 2\,000$
8. (17 баллов) $M = 2$
9. (12 баллов) Без дополнительных ограничений.

Sample Grader

Грейдер читает входные данные в следующем формате:

- строка 1: N M
- строка 2: $P[0]$ $P[1]$ \dots $P[N - 1]$
- строка 3: $C[0]$ $C[1]$ \dots $C[N - 1]$

Пусть $b[0]$, $b[1]$, \dots обозначает массив, который вернула функция `beechtree`. Грейдер выводит ответ в единственной строке в следующем формате:

- строка 1: $b[0]$ $b[1]$ \dots