درخت راش

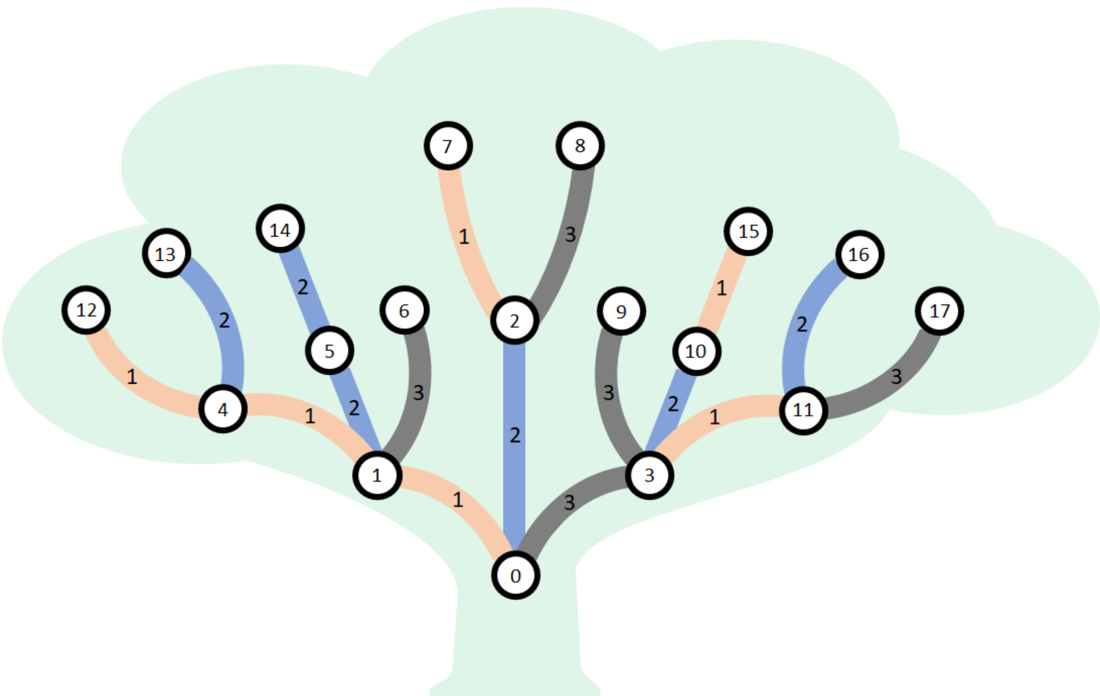
جنگل *ویتیم* جنگلی مشهور پر از درختان رنگارنگ است. یکی از مسن‌ترین و بلندترین درختان راش، *اوس وِزیر* نام دارد.

درخت *اوس وزیر* را می توان به شکل یک مجموعه از N راس و $N-1$ یال مدل کرد. **راس‌ها** از 0 تا $N-1$ شماره‌گذاری شده‌اند و **یال‌ها** از 1 تا $N-1$ شماره‌گذاری شده اند. هر یال دو راس متفاوت را به هم متصل می‌کند. به طور خاص، یال i $(1 \le i < N)$ راس i را به راس $P[i]$ متصل می‌کند، که $0 \le P[i] < i$. راس i، **فرزند** راس $P[i]$ خطاب می‌شود.

هر یال یک رنگ دارد. در کل M رنگ متفاوت داریم که از 1 تا M شماره‌گذاری شده‌اند. رنگ یال i، $C[i]$ است. ممکن است یال‌های متفاوت رنگ یکسانی داشته باشند.

دقت کنید در تعریف بالا، حالت $i=0$ به یک یال از درخت اشاره نمی‌کند. برای راحتی، $P[0]=-1$ و $C[0]=0$ تعریف می‌کنیم.

برای مثال، فرض کنید *اوس وزیر* $N=18$ راس و $M=3$ رنگِ ممکن برای یال دارد، و 17 یال با اتصالات $P=[-1,0,0,0,1,1,1,2,2,3,3,3,4,4,5,10,11,11]$ و رنگ‌های $C=[0,1,2,3,1,2,3,1,3,3,2,1,1,2,2,1,2,3]$. این درخت در تصویر زیر نمایش داده شده است:



آرپد یک جنگل‌بان مستعد است که علاقه دارد قسمت‌های بخصوصی از درخت به نام **زیردرخت** ها را مطالعه کند. به ازای هر r که $0 \le r < N$، زیردرخت راس r که با $T(r)$ نشان داده می‌شود، یک مجموعه از رئوس است. که شرایط زیر را دارد:

- راس r عضوی از $T(r)$ است.
- هر بار که راسی مانند x عضوی از $T(r)$ می‌شود، تمام فرزندهای x نیز عضوی از $T(r)$ می‌شوند.
- راس دیگری عضو $T(r)$ نمی‌شود.

سایز مجموعه‌ی $T(r)$ را با $|T(r)|$ نشان می‌دهیم.

ارپد به تازگی یک ویژگی پیچیده اما جالب راجع به زیردرخت را کشف کرده است. اکتشاف *ارپد* شامل میزان زیادی بازی با قلم و کاغذ می‌شد، و او فکر می‌کند ممکن است شما نیز برای درک آن نیاز به این کار داشته باشید. همچنین او به شما چندین مثال نشان می‌دهد که می‌توانید تحلیل کنید.

فرض کنید یک راس ثابت r و یک جایگشت $v_0, v_1, \ldots, v_{|T(r)|-1}$ از رئوس داخل $T(r)$ داریم.

به ازای هر i که $1 \leq i < |T(r)|$، $f(i)$ را تعریف می‌کنیم تعداد بارهایی که رنگ $C[v_i]$ در دنباله‌ی روبه‌رو که $i-1$ عضو دارد تکرار می‌شود: $C[v_1], C[v_2], \ldots, C[v_{i-1}]$.

(دقت کنید که $f(1)$ همیشه 0 است چون دنباله‌ی رنگ‌های در تعریفش تهی است.)

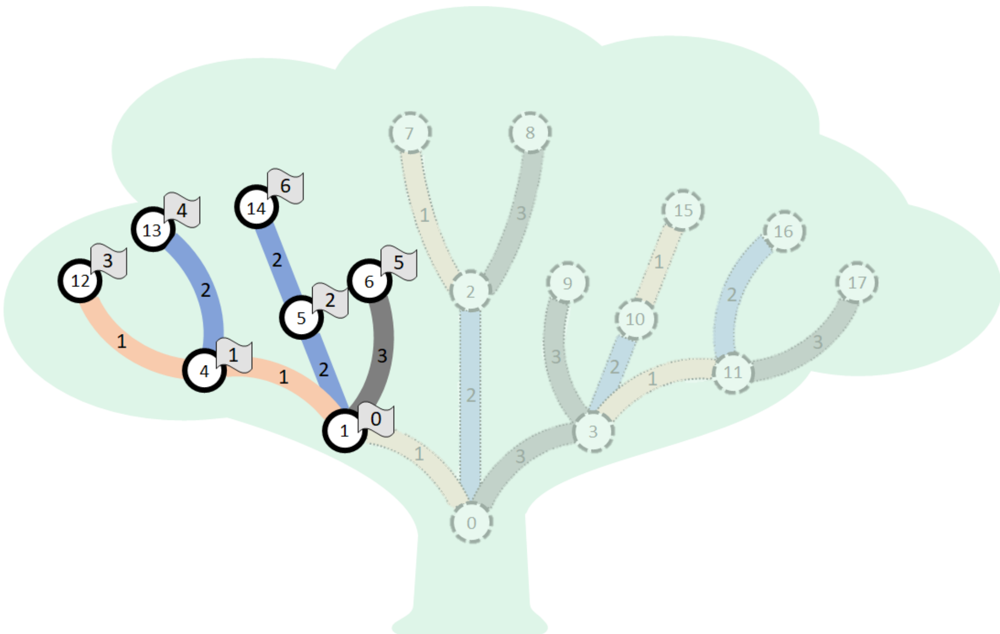
جایگشت $v_0, v_1, \ldots, v_{|T(r)|-1}$ یک **جایگشت زیبا** است اگر و تنها اگر تمام شرایط زیر برقرار باشند:

- $v_0 = r$.
- به ازای هر i که $1 \leq i < |T(r)|$، پدر راس v_i راس $v_{f(i)}$ باشد.

به ازای هر r که $0 \leq r < N$، زیردرخت $T(r)$ یک **زیردرخت زیبا** است اگر و تنها اگر جایگشتی زیبا از رئوس داخل $T(r)$ وجود داشته باشد. دقت کنید که طبق تعریف هر زیردرخت که تنها شامل یک راس می‌شود زیبا است.

درخت بالا را در نظر بگیرید. می‌توان نشان داد که زیردرخت‌های $T(0)$ و $T(3)$ از این درخت زیبا نیستند. زیردرخت $T(14)$ زیبا است، زیرا تنها یک راس دارد. در ادامه نشان می‌دهیم که زیردرخت $T(1)$ نیز زیبا است.

دنباله‌ی شامل اعداد متمایز $[v_0, v_1, v_2, v_3, v_4, v_5, v_6] = [1, 4, 5, 12, 13, 6, 14]$ را در نظر بگیرید. این دنباله یک جایگشت از رئوس داخل $T(1)$ است. تصویر زیر این جایگشت را به تصویر می‌کشد. برچسب‌های چسبیده به هر راس اندیس آن راس در جایگشت است.



به وضوح، دنباله‌ی بالا یک جایگشت از رئوس داخل $T(1)$ است. حالا نشان می‌دهیم که *زیبا* است.

- $v_0 = 1$.
- $f(1) = 0$ زیرا عدد $C[v_1] = C[4] = 1$ در دنباله‌ی $[]$ ، 0 بار تکرار شده است.
  - متقابلا، پدر v_1 ، v_0 است. یعنی پدر راس 4 راس 1 است. ($P[4] = 1$)
- $f(2) = 0$ زیرا عدد $C[v_2] = C[5] = 2$ در دنباله‌ی $[1]$ ، 0 بار تکرار شده است.
  - متقابلا، پدر v_2 ، v_0 است. یعنی پدر راس 5 راس 1 است.
- $f(3) = 1$ زیرا عدد $C[v_3] = C[12] = 1$ در دنباله‌ی $[1, 2]$ ، 1 بار تکرار شده است.
  - متقابلا، پدر v_3 ، v_1 است. یعنی پدر راس 12 راس 4 است.
- $f(4) = 1$ زیرا عدد $C[v_4] = C[13] = 2$ در دنباله‌ی $[1, 2, 1]$ ، 1 بار تکرار شده است.
  - متقابلا، پدر v_4 ، v_1 است. یعنی پدر راس 13 راس 4 است.
- $f(5) = 0$ زیرا عدد $C[v_5] = C[6] = 3$ در دنباله‌ی $[1, 2, 1, 2]$ ، 0 بار تکرار شده است.
  - متقابلا، پدر v_5 ، v_0 است. یعنی پدر راس 6 راس 1 است.
- $f(6) = 2$ زیرا عدد $C[v_6] = C[14] = 2$ در دنباله‌ی $[1, 2, 1, 2, 3]$ ، 2 بار تکرار شده است.
  - متقابلا، پدر v_6 ، v_2 است. یعنی پدر راس 14 راس 5 است.

حالا که یک جایگشت زیبا از راس‌های درون $T(1)$ پیدا کردیم، زیردرخت $T(1)$ براستی زیبا است.

شما باید به *ارید* کمک کنید تا برای هر زیردرخت از *اوس وزیر* تعیین کند که زیبا هست یا خیر.

Implementation Details

You should implement the following procedure.

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

- N: the number of nodes in the tree.
- M: the number of possible edge colors.
- P, C: arrays of length N describing the edges of the tree.
- This procedure should return an array b of length N. For each r such that $0 \leq r < N$, $b[r]$ should be 1 if $T(r)$ is beautiful, and 0 otherwise.
- This procedure is called exactly once for each test case.

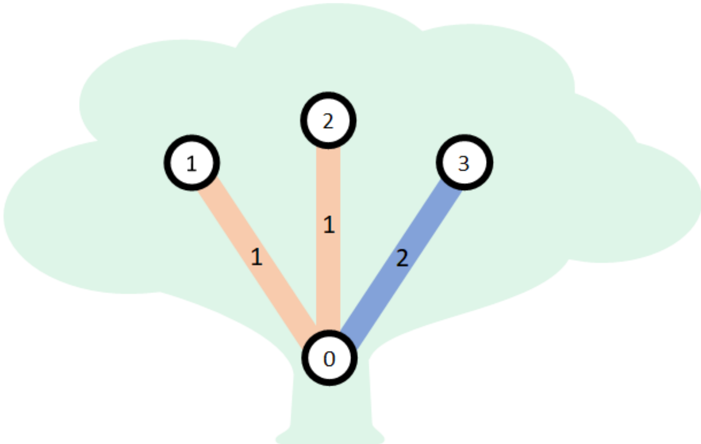
Examples

Example 1

Consider the following call:

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

The tree is displayed in the following figure:



$T(1)$, $T(2)$ and $T(3)$ each consist of a single node and are therefore beautiful. $T(0)$ is not beautiful. Therefore, the procedure should return $[0, 1, 1, 1]$.

Example 2

Consider the following call:

```
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])
```

This example is illustrated in the task description above.

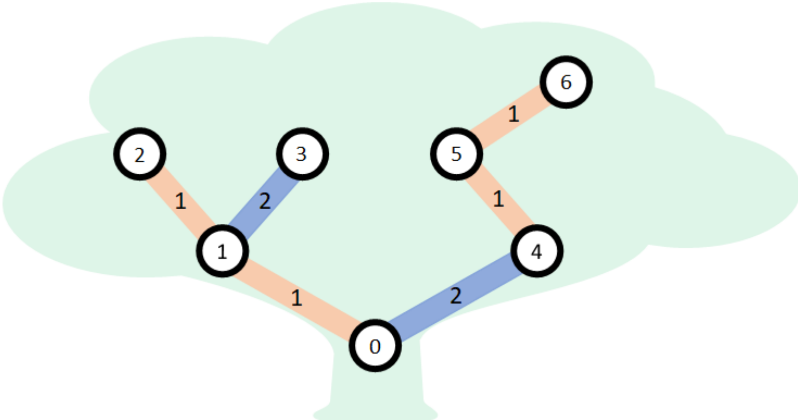
The procedure should return $[0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$.

Example 3

Consider the following call:

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

This example is illustrated in the following figure.



.$[0,1,1,1,1,1,1]$ is the only subtree that is not beautiful. The procedure should return $T(0)$

Constraints

- $3 \leq N \leq 200\,000$
- $2 \leq M \leq 200\,000$
- $(1 \leq i < N$ for each i such that$)$ $0 \leq P[i] < i$
- $(1 \leq i < N$ for each i such that$)$ $1 \leq C[i] \leq M$
- $C[0] = 0$ and $P[0] = -1$

Subtasks

1. $M \leq 500$ points) $N \leq 8$ and 9) .1
2. ,$1 \leq i < N$ points) Edge i connects node i to node $i-1$. That is, for each i such that 5) .2
.$P[i] = i - 1$
3. points) Each node other than node 0 is either connected to node 0, or is connected to a 9) .3
node which is connected to node 0. That is, for each v such that $1 \leq v < N$, either $P[v] = 0$
.$P[P[v]] = 0$ or
4. .c points) For each c such that $1 \leq c \leq M$, there are at most two edges of color 8) .4
5. $M \leq 500$ points) $N \leq 200$ and 14) .5
6. $M = 2$ points) $N \leq 2\,000$ and 14) .6
7. $N \leq 2\,000$ (points 12) .7
8. $M = 2$ (points 17) .8
9. .points) No additional constraints 12) .9

Sample Grader

:The sample grader reads the input in the following format

- $N\ M$:1 line
- $P[0]\ P[1]\ \ldots\ P[N-1]$:2 line
- $C[0]\ C[1]\ \ldots\ C[N-1]$:3 line

Let $b[0],\ b[1],\ \ldots$ denote the elements of the array returned by `beechtree`. The sample grader :prints your answer in a single line, in the following format

- line 1: $b[0]\ b[1]\ \ldots$