Buche

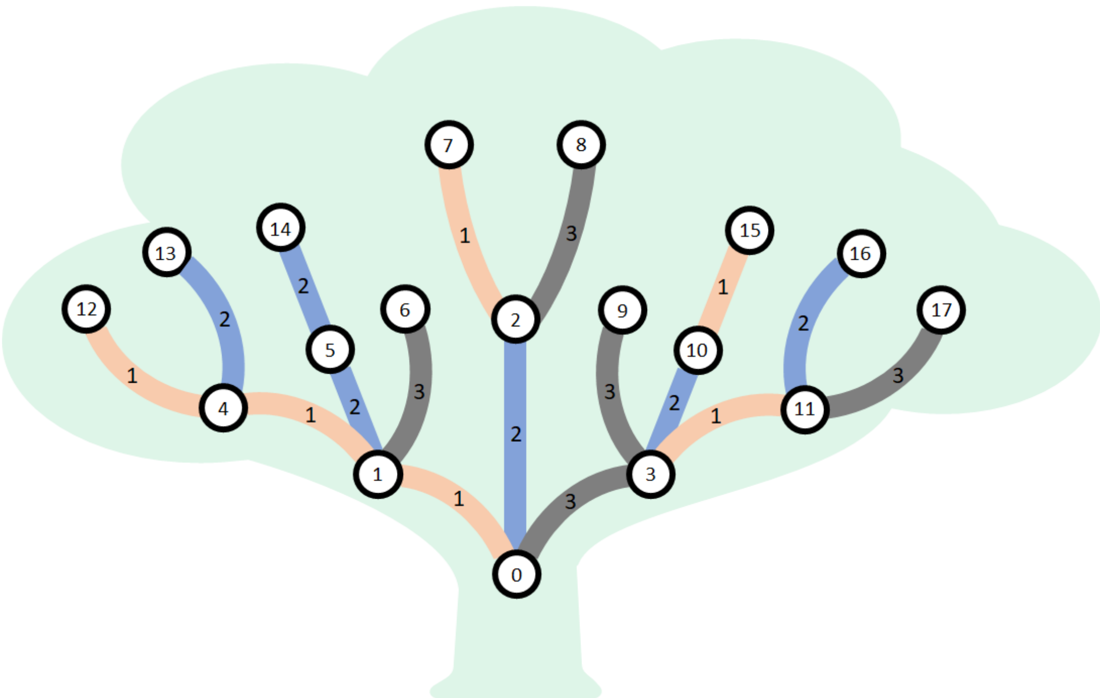
Die Vétyem-Wälder sind ein berühmtes Waldgebiet mit vielen bunten Bäumen. Eine der ältesten und höchsten Buchen heißt Ős Vezér.

Der Baum Ős Vezér kann als eine Menge von N **Knoten** und $N-1$ **Kanten** modelliert werden. Die Knoten sind von 0 bis $N-1$ nummeriert, und die Kanten sind von 1 bis $N-1$ nummeriert. Jede Kante verbindet zwei verschiedene Knoten des Baums. Genauer gesagt, verbindet Kante i ($1 \leq i < N$) Knoten i mit Knoten $P[i]$ ($0 \leq P[i] < i$). $P[i]$ heißt auch **Elternknoten** von i, und Knoten i heißt **Kind** von $P[i]$.

Jede Kante hat eine Farbe. Es gibt M mögliche Kantenfarben, nummeriert von 1 bis M. Die Farbe der Kante i ist $C[i]$. Verschiedene Kanten können die gleiche Farbe haben.

Für $i = 0$ gibt es nach der Definition keinen Elternknoten. Der Einfachheit halber sei $P[0] = -1$ und $C[0] = 0$.

Ein Beispiel: Ős Vezér hat $N = 18$ Knoten, mit $M = 3$ möglichen Kantenfarben. Die 17 Kanten und deren Farben sind gegeben durch $P = [-1, 0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 3, 4, 4, 5, 10, 11, 11]$ und $C = [0, 1, 2, 3, 1, 2, 3, 1, 3, 3, 2, 1, 1, 2, 2, 1, 2, 3]$. Der Baum ist hier abgebildet:



Árpád ist ein talentierter Förster, der gerne besondere Teile eines Baumes studiert: die **Teilbäume**. Der Teilbaum $T(r)$ von Knoten r ist (für alle $0 \le r < N$) eine Menge von Knoten mit den folgenden Eigenschaften:

- Knoten r gehört zu $T(r)$.
- Wenn ein Knoten x zu $T(r)$ gehört, dann auch alle Kinder von x.
- Kein anderer Knoten gehört zu $T(r)$.

Die Größe der Menge $T(r)$ wird mit $|T(r)|$ bezeichnet.

Árpád hat kürzlich eine sehr interessante, aber komplizierte Eigenschaft einiger Teilbäume entdeckt. Dazu hat er lange mit Stift und Papier probiert. Er vermutet, dass auch du das tun solltest, um seine Entdeckung zu verstehen. Er wird dir auch einige Beispiele zeigen, die du analysieren kannst.

Im Folgenden gehen wir von einem festen r und einer Permutation $v_0, v_1, \ldots, v_{|T(r)|-1}$ der Knoten im Teilbaum $T(r)$ aus.

Für jedes i mit $1 \le i < |T(r)|$, sei $f(i)$ die Anzahl, wie oft die Farbe $C[v_i]$ in der Folge der Farben $C[v_1], C[v_2], \ldots, C[v_{i-1}]$ vorkommt. ($f(1)$ ist immer 0, da diese Folge für $i = 1$ leer ist.)

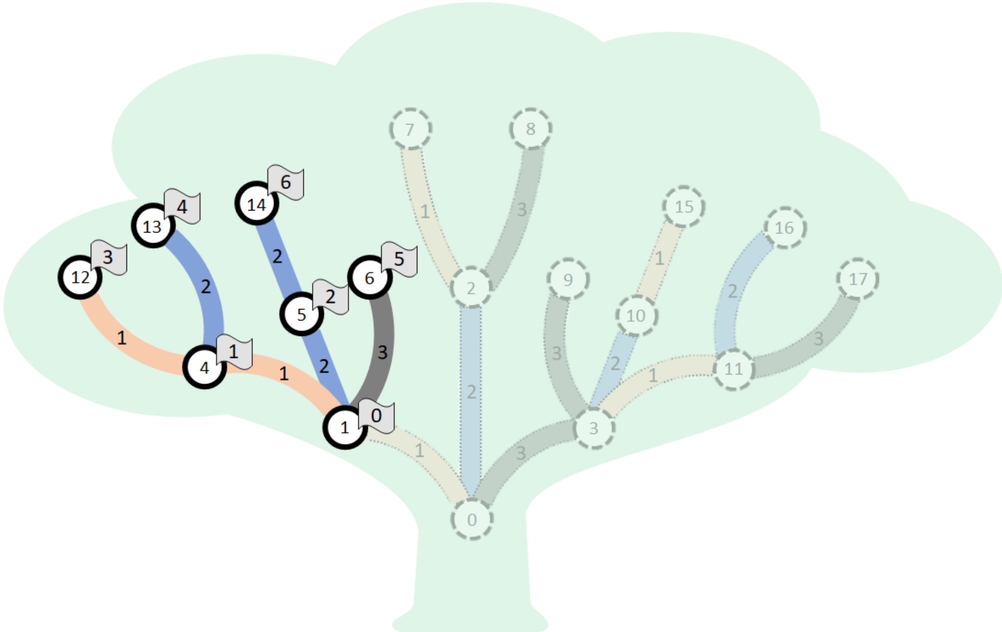
Die Permutation der Knoten(nummern) $v_0, v_1, \ldots, v_{|T(r)|-1}$ ist **wunderschön**, genau dann wenn die beiden folgenden Eigenschaften erfüllt sind:

- $v_0 = r$.
- Für jedes i mit $1 \le i < |T(r)|$ ist der Knoten $v_{f(i)}$ der Elternknoten von v_i.

Einen Teilbaum $T(r)$ ($0 \le r < N$) nennen wir **wunderschön**, genau dann wenn eine wunderschöne Permutation der Knoten in $T(r)$ existiert. Nach dieser Definition ist jeder Teilbaum wunderschön, der nur einen einzigen Knoten enthält.

Betrachte den Beispielbaum oben. Die Teilbäume $T(0)$ und $T(3)$ dieses Baumes sind nicht wunderschön, wie man zeigen kann. Der Teilbaum $T(14)$ ist wunderschön, da er nur einen einzigen Knoten enthält. Nun werden wir zeigen, dass Teilbaum $T(1)$ auch wunderschön ist.

Betrachte die Folge von verschiedenen ganzen Zahlen $[v_0, v_1, v_2, v_3, v_4, v_5, v_6] = [1, 4, 5, 12, 13, 6, 14]$. Diese Folge ist eine Permutation der Knoten in $T(1)$; sie ist im folgenden Bild dargestellt. Die Etiketten an den Knoten zeigen die Indizes der jeweiligen Knoten in dieser Folge.



Offensichtlich ist die obige Folge eine Permutation der Knoten in $T(1)$. Wir zeigen nun, dass sie auch *wunderschön* ist.

- $v_0 = 1$.
- $f(1) = 0$, denn $C[v_1] = C[4] = 1$ kommt 0 Mal in der Folge $[]$ vor.
  - Entsprechend ist v_0 der Elternknoten von v_1 (d.h. 1 ist der Elternknoten von 4).
- $f(2) = 0$, denn $C[v_2] = C[5] = 2$ kommt 0 Mal in der Folge $[1]$ vor.
  - Entsprechend ist v_0 der Elternknoten von v_2 (d.h. 1 ist der Elternknoten von 5).
- $f(3) = 1$, denn $C[v_3] = C[12] = 1$ kommt 1 Mal in der Folge $[1, 2]$ vor.
  - Entsprechend ist v_1 der Elternknoten von v_3 (d.h. 4 ist der Elternknoten von 12).
- $f(4) = 1$, denn $C[v_4] = C[13] = 2$ kommt 1 Mal in der Folge $[1, 2, 1]$ vor.
  - Entsprechend ist v_1 der Elternknoten von v_4 (d.h. 4 ist der Elternknoten von 13).
- $f(5) = 0$, denn $C[v_5] = C[6] = 3$ kommt 0 Mal in der Folge $[1, 2, 1, 2]$ vor.
  - Entsprechend ist v_0 der Elternknoten von v_5 (d.h. 1 ist der Elternknoten von 6).
- $f(6) = 2$, denn $C[v_6] = C[14] = 2$ kommt 2 Mal in der Folge $[1, 2, 1, 2, 3]$ vor.
  - Entsprechend ist v_2 der Elternknoten von v_6 (d.h. 5 ist der Elternknoten von 14).

Da wir eine wunderschöne Permutation der Knoten in $T(1)$ gefunden haben, ist der Teilbaum $T(1)$ in der Tat wunderschön.

Hilf Árpád, für jeden Teilbaum von Ős Vezér zu entscheiden, ob er wunderschön ist.

Implementierungsdetails

Implementiere die folgende Funktion:

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

- N: die Anzahl an Knoten.

- M: die Anzahl an möglichen Kantenfarben.
- P, C: Arrays der Länge N, welche die Kanten des Baumes beschreiben.
- Diese Funktion soll ein Array b der Länge N zurückgeben. Für jedes r mit $0 \leq r < N$ sollte $b[r]$ gleich 1 sein falls $T(r)$ wunderschön ist, und sonst gleich 0.
- Diese Funktion wird für jeden Testfall genau einmal aufgerufen.

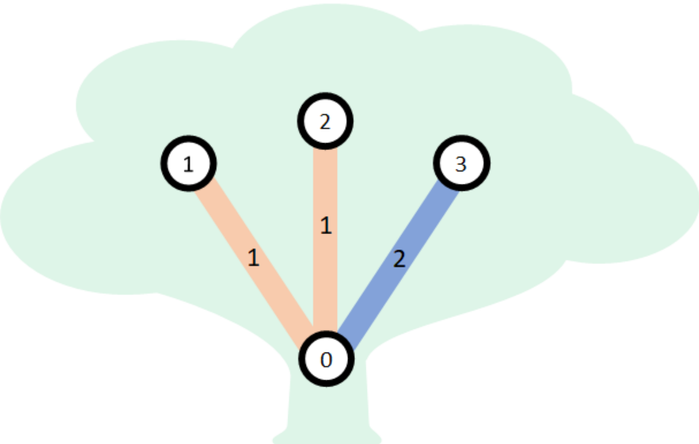
Beispiele

Beispiel 1

Betrachte den folgenden Funktionsaufruf:

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

Dieses Beispiel entspricht dem folgenden Baum:



$T(1)$, $T(2)$, und $T(3)$ bestehen jeweils aus einem einzigen Knoten, und sind somit wunderschön. $T(0)$ ist nicht wunderschön. Deswegen sollte die Funktion $[0, 1, 1, 1]$ zurückgeben.

Beispiel 2

Betrachte den folgenden Funktionsaufruf:

```
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])
```

Dieses Beispiel entspricht dem Baum der Aufgabenstellung.

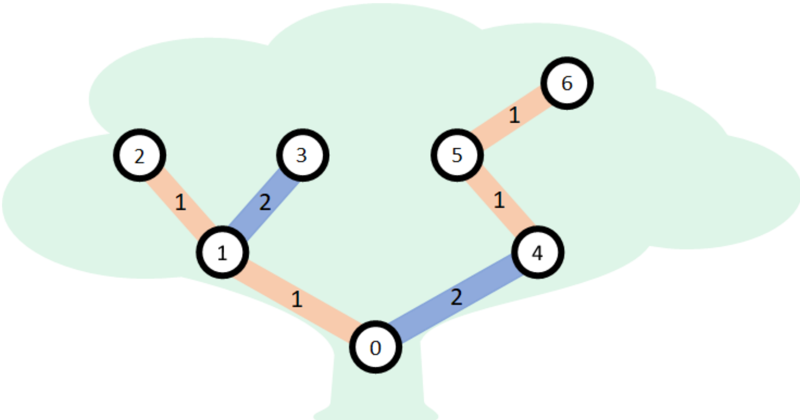
Die Funktion soll $[0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$ zurückgeben.

Beispiel 3

Betrachte den folgenden Funktionsaufruf:

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

Dieses Beispiel entspricht dem folgenden Baum:



$T(0)$ ist der einzige Teilbaum, der nicht wunderschön ist. Die Funktion soll $[0, 1, 1, 1, 1, 1, 1]$ zurückgeben.

Beschränkungen

- $3 \leq N \leq 200\,000$
- $2 \leq M \leq 200\,000$
- $0 \leq P[i] < i$ ($1 \leq i < N$)
- $1 \leq C[i] \leq M$ ($1 \leq i < N$)
- $P[0] = -1$ und $C[0] = 0$

Teilaufgaben

1. (9 Punkte) $N \leq 8$ und $M \leq 500$
2. (5 Punkte) Kante i verbindet Knoten i mit Knoten $i - 1$. Das heißt, für jedes $1 \leq i < N$, $P[i] = i - 1$.
3. (9 Punkte) Jeder Knoten, der nicht Knoten 0 ist, ist entweder mit Knoten 0 verbunden oder mit einem Knoten verbunden, der mit Knoten 0 verbunden ist. Das heißt, für jedes $1 \leq i < N$ gilt entweder $P[i] = 0$ oder $P[P[i]] = 0$.
4. (8 Punkte) Für jedes $1 \leq c \leq M$ gibt es maximal zwei Kanten mit Farbe c.
5. (14 Punkte) $N \leq 200$ und $M \leq 500$
6. (14 Punkte) $N \leq 2\,000$ und $M = 2$
7. (12 Punkte) $N \leq 2\,000$
8. (17 Punkte) $M = 2$
9. (12 Punkte) Keine weiteren Beschränkungen.

Beispielgrader

Der Beispielgrader liest die Eingabe im folgenden Format:

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

Seien $b[0], \ b[1], \ \ldots$ die Elemente des Arrays, das `beechtree` zurückgibt. Der Beispielgrader gibt deine Antwort auf einer einzelnen Zeile aus, im folgenden Format:

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