Beech Tree

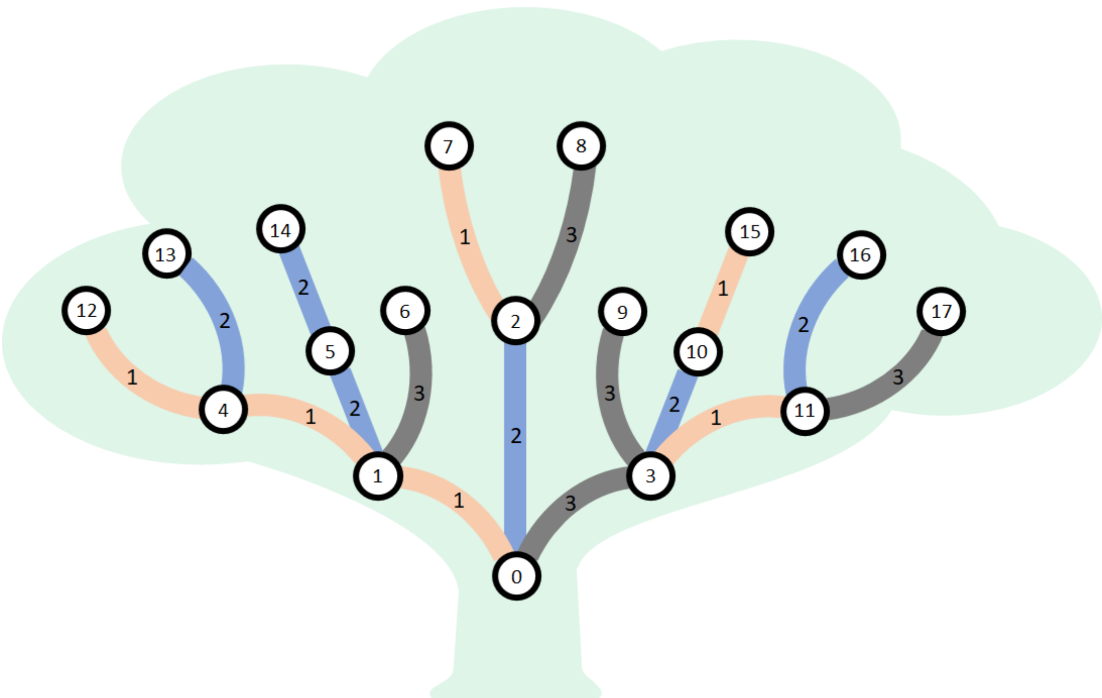
Het Vétyem Bos is een beroemd bos met vele kleurrijke bomen. Één van de oudste en grootste beukenboom wordt Ős Vezér genoemd.

De boom Ős Vezér kun je modelleren als een verzameling van N **knopen** en $N-1$ **takken**. De knopen zijn genummerd van 0 tot en met $N-1$, en de takken van 1 tot en met $N-1$. Elke tak verbindt twee verschillende knopen van de boom. Om precies te zijn, verbindt tak i ($1 \leq i < N$) knoop i met knoop $P[i]$, waarbij $0 \leq P[i] < i$. Knoop $P[i]$ heet de **ouder** van knoop i en knoop i heet een **kind** van $P(i)$.

Elke tak heeft een kleur. Er zijn M mogelijke kleuren voor de takken, genummerd van 1 tot en met M. De kleur van tak i is $C[i]$. Verschillende takken mogen dezelfde kleur hebben.

Merk op dat $i = 0$ in de definitie hierboven niet overeenkomt met een tak in de boom. Voor het gemak is $P[0] = -1$ en $C[0] = 0$.

Bijvoorbeeld, stel Ős Vezér heeft $N = 18$ knopen en $M = 3$ mogelijke kleuren van de takken, met 17 takken beschreven door de verbindingen $P = [-1, 0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 3, 4, 4, 5, 10, 11, 11]$ en kleuren $C = [0, 1, 2, 3, 1, 2, 3, 1, 3, 3, 2, 1, 1, 2, 2, 1, 2, 3]$. De boom is te zien in het figuur hieronder.



Árpád is een getalenteerd bosbouwkundige, die graag stukken van de boom bestudeerd, ook wel **deelbomen** genoemd. Voor elke r waarvoor $0 \le r < N$, is de deelboom van knoop r de verzameling knopen $T(r)$. met de volgende eigenschappen:

- Knoop r behoort tot $T(r)$.
- Als een knoop x in $T(r)$ zit, dan horen ook alle kinderen van x bij $T(r)$.
- Geen enkele andere knoop maakt deel uit van $T(r)$.

De grootte van de verzameling $T(r)$ wordt genoteerd met $|T(r)|$.

Árpád ontdekte onlangs een interessante eigenschap van deelbomen. Voor zijn ontdekking had Árpád's heel wat met pen en papier zitten spelen, en hij verwacht dat je hetzelfde zult moeten doen om te begrijpoen waar het over gaat. Hij zal je een heleboel voorbeelden geven die je in detail kunt bestuderen.

Stel dat we een vaste i hebben en een permutatie v_0, $v_1 \ldots v_{|T(r)|-1}$ van de knopen in de deelboom $T(r)$.

Voor elke i waarvoor $0 \le i < |T(r)|$, zit knoop v_i in $T(r)$.

Voor elke i zodat $1 \le i < |T(r)|$, laat $f(i)$ het aantal keren zijn dat de kleur $C[v_i]$ voorkomt in de volgende rij van $i-1$ kleuren $C[v_1], C[v_2], \ldots, C[v_{i-1}]$.

(Let op dat $f(1)$ altijd 0 is omdat de rij met kleuren per definitie leeg is.)

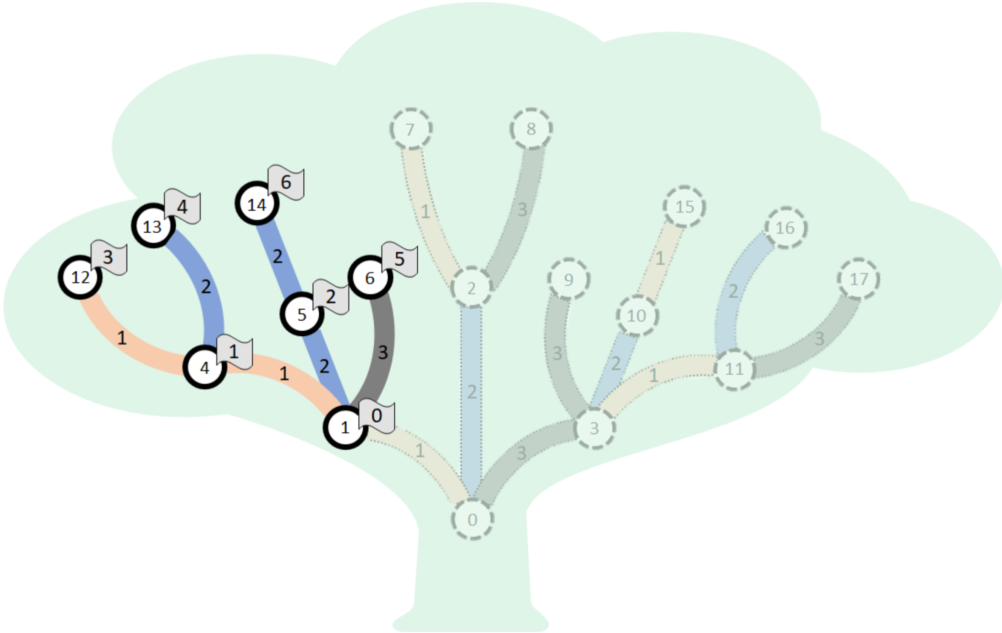
De permutatie $v_0, v_1, \ldots, v_{|T(r)|-1}$ is **een prachtige permutatie** dan en slechts dan als alle volgende eigenschappen gelden:

- $v_0 = r$.
- Voor elke i zodat $1 \le i < |T(r)|$, is de ouder van knoop v_i knoop $v_{f(i)}$.

Voor elke r met $0 \le r < N$, is $T(r)$ een **prachtige deelboom** dan en slechts dan als er een prachtige permutatie bestaat van de knopen in $T(r)$. Merk op dat volgens de definitie elke deelboom van een enkele knoop prachtig is.

Bekijk de voorbeeldboom hierboven. Je kunt laten zien dat de deelbomen $T(0)$ en $T(3)$ niet prachtig zijn. De deelboom $T(14)$ is prachtig, omdat het maar één knoop heeft. Later wordt uitgelegd dat $T(1)$ ook prachtig is.

Bekijk de rij van verschillende gehele getallen $[v_0, v_1, v_2, v_3, v_4, v_5, v_6] = [1, 4, 5, 12, 13, 6, 14]$. Deze rij is een permutatie van de knopen in $T(1)$. In de onderstaande figuur zijn de knopen indices die die bij elke knoop in deze rij staat als nummer in het vlaggetje bij de knoop.



We gaan nu na dat dit een *prachtige permutatie* is.

- $v_0 = 1$.
- $f(1) = 0$ want $C[v_1] = C[4] = 1$ komt 0 keer voor in de rij $[]$.
  - Daarmee stemt overeen dat de ouder van v_1 is v_0. Oftewel, de ouder van knoop 4 is knoop 1. (Formeel, $P[4] = 1$)
- $f(2) = 0$ want $C[v_2] = C[5] = 2$ komt 0 keer voor in de rij $[1]$.
  - Daarmee stemt overeen dat de ouder van v_2 is v_0. (dwz de ouder van knoop 5 is knoop 1.)
- $f(3) = 1$ want $C[v_3] = C[12] = 1$ komt 1 keer voor in de rij $[1, 2]$.
  - Daarmee stemt overeen dat de ouder van v_3 is v_1. (dwz de ouder van knoop 12 is knoop 4.)
- $f(4) = 1$ want $C[v_4] = C[13] = 2$ komt 1 keer voor in de rij $[1, 2, 1]$.
  - Daarmee stemt overeen dat de ouder van v_4 is v_1. (dwz de ouder van knoop 13 is knoop 4.)
- $f(5) = 0$ want $C[v_5] = C[6] = 3$ komt 0 keer voor in de rij $[1, 2, 1, 2]$.
  - Daarmee stemt overeen dat de ouder van v_5 is v_0. (dwz de ouder van knoop 6 is knoop 1.)
- $f(6) = 2$ want $C[v_6] = C[14] = 2$ komt 2 keer voor in de rij $[1, 2, 1, 2, 3]$.
  - Daarmee stemt overeen dat de ouder van v_6 is v_2. (dwz de ouder van 14 is 5.)

We kunnen dus een *prachtige permutatie* vinden van de knopen in $T(1)$, de deelboom $T(1)$ is inderdaad *prachtig*.

Jouw taak is om Árpád te helpen beslissen of elke deelboom van Ős Vezér prachtig is of niet.

Implementatiedetails

Je moet de volgende functie implementeren.

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

- N: het aantal knopen in de boom.
- M: het aantal mogelijke kleuren voor de takken.
- P, C: rijen van lengte N die de takken van de boom beschrijven.
- De functie moet een rij b teruggeven van lengte N. Voor elke r waarvoor $0 \leq r < N$, moet $b[r]$ gelijk zijn aan 1 als $T(r)$ prachtig is, en anders 0.
- Deze functie wordt precies één keer in een testcase aangeroepen.

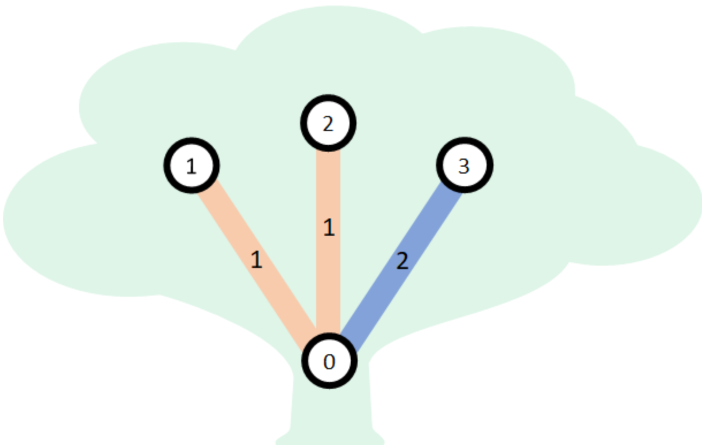
Voorbeelden

Voorbeeld 1

Bekijk de volgende aanroep:

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

De boom is te zien in onderstaand figuur.



$T(1)$, $T(2)$, en $T(3)$ bestaan alle drie uit één knoop en zijn daardoor prachtig. $T(0)$ is niet prachtig. Daarom moet de functie $[0, 1, 1, 1]$ teruggeven.

Voorbeeld 2

Bekijk de volgende aanroep:

```
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])
```

Dit voorbeeld is in het begin van de opgave al beschreven.

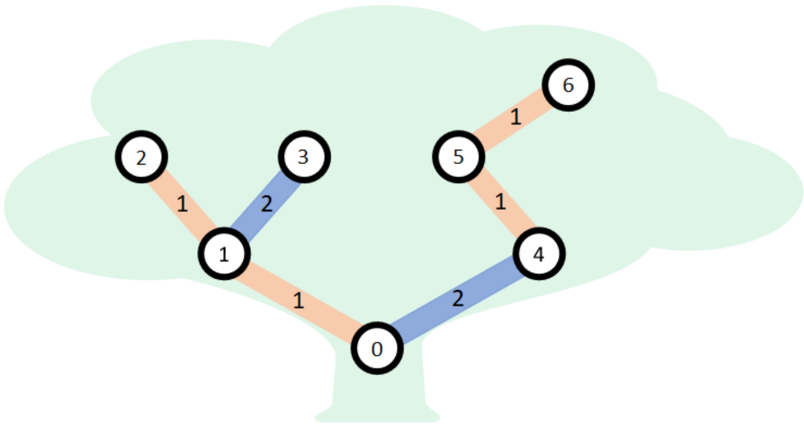
De functie moet $[0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$ teruggeven.

Voorbeeld 3

Bekijk de volgende aanroep:

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

Dit voorbeeld is te zien in onderstaand figuur.



$T(0)$ is de enige deelboom die niet prachtig is. De functie moet $[0, 1, 1, 1, 1, 1, 1]$ teruggeven.

Constraints

- $3 \leq N \leq 200\,000$
- $2 \leq M \leq 200\,000$
- $0 \leq P[i] < v$ (voor elke i zodat $1 \leq i < N$)
- $1 \leq C[i] \leq M$ (voor elke i zodat $1 \leq i < N$)
- $P[0] = -1$ en $C[0] = 0$

Subtasks

1. (9 punten) $N \leq 8$ en $M \leq 500$
2. (5 punten) Tak i verbindt knoop i met knoop $i - 1$. Oftewel, voor elke i waarvoor $1 \leq i < N$, geldt $P[i] = i - 1$.
3. (9 punten) Elke knoop behalve knoop 0 is ofwel verbonden met knoop 0 of verbonden met een knoop die verbonden is met knoop 0. Dat wil zeggen, voor elke i waarvoor $1 \leq i < N$, geldt ofwel $P[i] = 0$ of $P[P[i]] = 0$.
4. (8 punten) Voor elke c waarvoor $1 \leq c \leq M$, zijn er hoogstens twee takken met kleur c.
5. (14 punten) $N \leq 200$ en $M \leq 500$
6. (14 punten) $N \leq 2\,000$ en $M = 2$
7. (12 punten) $N \leq 2\,000$
8. (17 punten) $M = 2$
9. (12 punten) Geen aanvullende randvoorwaarden.

Sample Grader

De sample grader leest de invoer in het volgende format:

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

Stel dat `beechtree` als resultaat de rij elementen $b[0], \ b[1], \ \ldots$ teruggeeft. Dan print de sample grader je antwooord op een enkele regel, in het volgende format:

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