Beech Tree

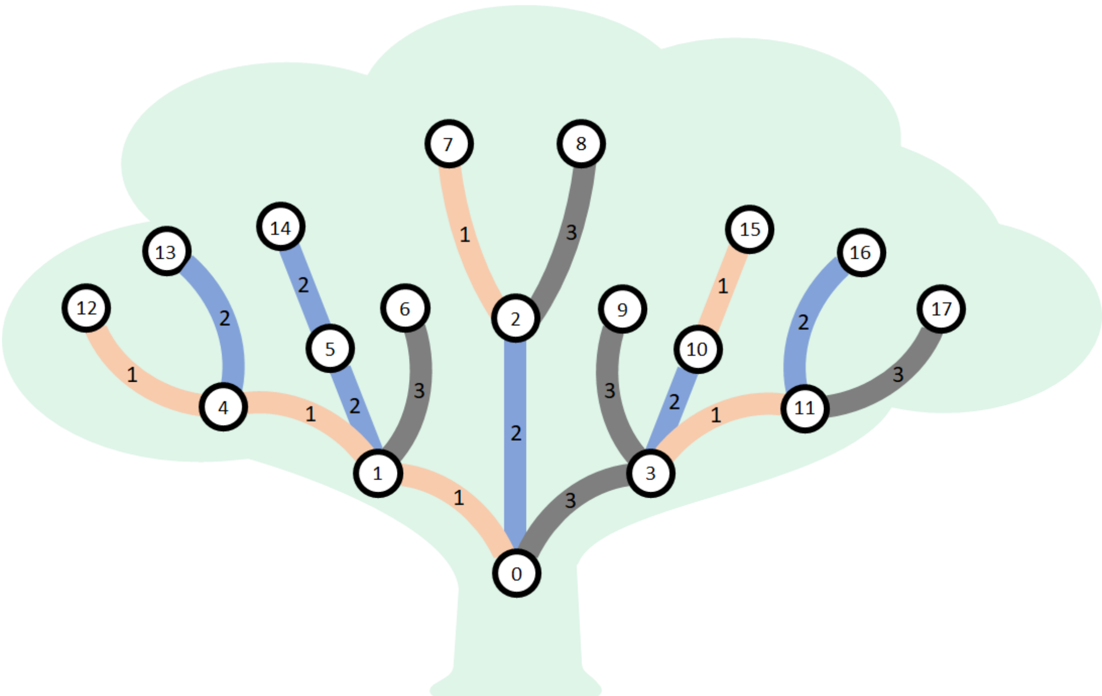
Vétyem Woods este o pădure vestită cu o mulțime de copaci colorați. Unul dintre cei mai vechi și înalți fagi se numește Ős Vezér.

Arborele Ős Vezér poate fi modelat ca o mulțime de N **noduri** și $N-1$ **muchii**. Nodurile sunt numerotate de la 0 la $N-1$, iar muchiile de la 1 la $N-1$. Fiecare muchie conectează două noduri ale arborelui. Mai exact, muchia i ($1 \leq i < N$) conectează nodul i cu nodul $P[i]$, unde $0 \leq P[i] < v$. Nodul $P[i]$ se numește **tatăl** nodului i, iar i este **fiu** al nodului $P[i]$.

Fiecare muchie are o culoare. Există M posibile culori numerotate de la 1 la M. Culoarea muchiei i este $C[i]$. Diferite muchii pot avea aceeași culoare.

De notat că în definiția de mai sus, cazul $i = 0$ nu corespunde unei muchii a arborelui. Pentru comoditate, vom considera $P[0] = -1$ și $C[0] = 0$.

De exemplu, presupunem că Ős Vezér are $N = 18$ noduri și $M = 3$ posibile culori ale muchiilor, cu 17 muchii descrise de conexiunile $P = [-1, 0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 3, 4, 4, 5, 10, 11, 11]$ și culorile $C = [0, 1, 2, 3, 1, 2, 3, 1, 3, 3, 2, 1, 1, 2, 2, 1, 2, 3]$. Arborele este prezentat în următoarea figură:



Árpád este un pădurar talentat căruia îi place să studieze părți specifice ale arborelui numite **subarbori**. Pentru fiecare r, $0 \leq r < N$, subarborele nodului r este setul $T(r)$ de noduri cu următoarele proprietăți:

- Nodul r aparține lui $T(r)$.
- Când un nod x aparține lui $T(r)$, toți fiii lui x aparține de asemenea lui $T(r)$.
- Niciun alt nod nu aparține lui $T(r)$.

Dimensiunea mulțimii $T(r)$ este notată prin $|T(r)|$.

Árpád a descoperit recent o proprietate complicată dar interesantă a subarborelui. Descoperirea lui Árpád a însemnat o mulțime de lucru cu hârtia și creionul și crede că și tu ai nevoie de același lucru pentru a înțelege. Îți va arăta și o mulțime de exemple pe care să le analizezi în amănunt.

Presupunem că avem o permutare fixată r și o permutare $v_0, v_1, \ldots, v_{|T(r)|-1}$ de noduri din subarborele $T(r)$.

Pentru fiecare i, astfel încât $1 \le i < |T(r)|$, fie $f(i)$ numărul de ori când culoarea $C[v_i]$ apare în următoarea secvență de $i-1$ culori: $C[v_1], C[v_2], \ldots, C[v_{i-1}]$.

(Observați că $f(1)$ este mereu 0 pentru că secvența de culori din definiția sa este vidă.)

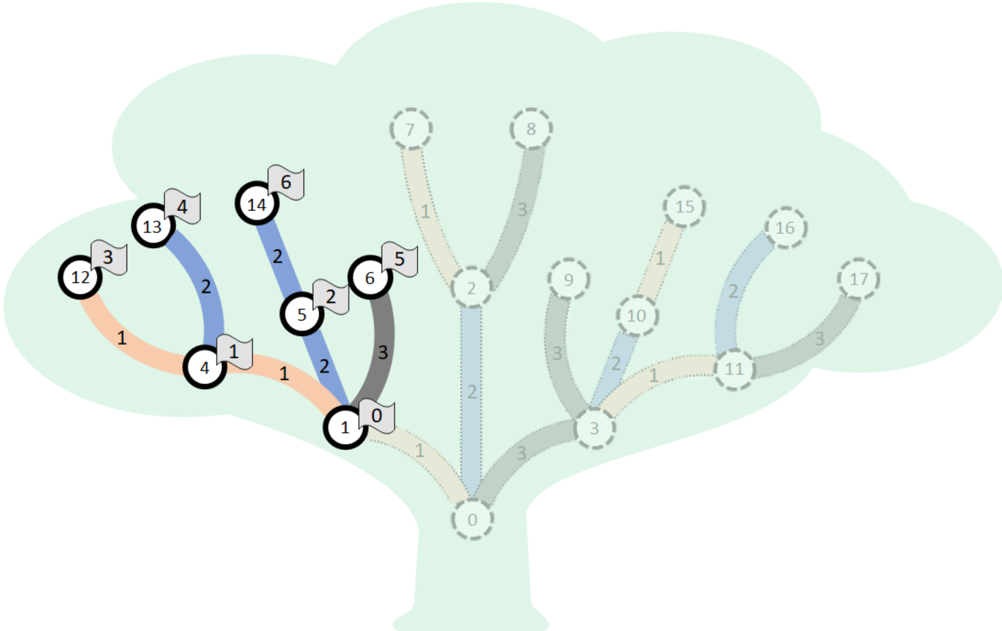
Permutarea $v_0, v_1, \ldots, v_{|T(r)|-1}$ este o **permutare frumoasă** dacă și numai dacă au loc proprietățile:

- $v_0 = r$.
- Pentru fiecare i cu $1 \le i < |T(r)|$, tatăl nodului v_i este nodul $v_{f(i)}$.

Pentru oricare r astfel încât $0 \le r < N$, subarborele $T(r)$ se numește un **subarbore frumos** dacă și numai dacă există o permutare frumoasă a nodurilor în $T(r)$. Observăm că conform definiției oricare subarbore ce conține un singur nod este frumos.

Să considerăm arborele exemplu de mai sus. Se poate arăta că subarborii $T(0)$ și $T(3)$ din acest arbore nu sunt frumoși. Subarborele $T(14)$ este frumos, deoarece constă dintr-un singur nod. Mai jos vom arăta că subarborele $T(1)$ de asemenea este frumos.

Să considerăm secvența de întregi distincți $[v_0, v_1, v_2, v_3, v_4, v_5, v_6] = [1, 4, 5, 12, 13, 6, 14]$. Această secvență este o permutare a nodurilor din $T(1)$. Permutarea de mai jos ilustrează o permutare. Etichetele atașate nodurilor sunt indicii la care acele noduri apar în permutare.



În mod clar, secvența de mai sus este o permutare a nodurilor din $T(1)$. Vom verifica dacă este și *frumoasă*.

- $v_0 = 1$.
- $f(1) = 0$ dacă $C[v_1] = C[4] = 1$ apare de 0 ori în secvența $[]$.
- Corespunzător, tatăl lui v_1 este v_0. Deci tatăl nodului 4 este nodul 1. (Formal, $P[4] = 1$.)
- $f(2) = 0$ dacă $C[v_2] = C[5] = 2$ apare de 0 ori în secvența $[1]$.
- Corespunzător, tatăl lui v_2 is v_0. Deci tatăl lui 5 este 1.
- $f(3) = 1$ dacă $C[v_3] = C[12] = 1$ apare de 1 ori în secvența $[1, 2]$.
- Corespunzător, tatăl lui v_3 is v_1. Deci tatăl lui 12 este 4.
- $f(4) = 1$ dacă $C[v_4] = C[13] = 2$ apare de 1 ori în secvența $[1, 2, 1]$.
- Corespunzător, tatăl lui v_4 is v_1. Deci tatăl lui 13 este 4.
- $f(5) = 0$ dacă $C[v_5] = C[6] = 3$ apare de 0 ori în secvența $[1, 2, 1, 2]$.
- Corespunzător, tatăl lui v_5 is v_0. Deci tatăl lui 6 este 1.
- $f(6) = 2$ since $C[v_6] = C[14] = 2$ apare de 2 ori în secvența $[1, 2, 1, 2, 3]$.
- Corespunzător, tatăl lui v_6 is v_2. Deci tatăl lui 14 este 5.

Putând găsi o *permutare frumoasă* în nodurile din $T(1)$, subarborele $T(1)$ este într-adevăr *subarbore frumos*.

Sarcina ta este să-l ajuți pe Árpád să decidă pentru fiecare subarbore din Ős Vezér dacă acesta este contorsionat.

Detalii de implementare

Urmează să implementezi următoarea procedură.

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

- N: numărul de noduri din arbore.
- M: numărul de culori posibile ale muchiilor.
- P, C: tablouri de dimensiunea N ce descriu muchiile din arbore.
- Această procedură va returna un tablou b de lungime N. Pentru fiecare r, $0 \leq r < N$, $b[r]$ trebuie să fie 1 dacă $T(r)$ este frumos, și 0 în caz contrar.
- Această procedură este apelată o singură dată pentru fiecare test.

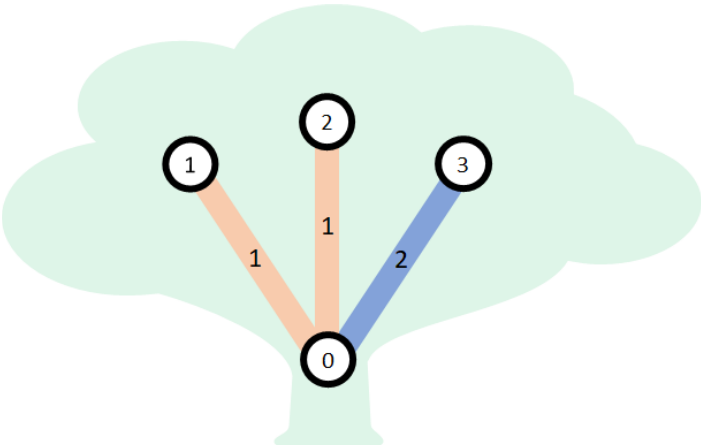
Exemple

Exemplul 1

Să considerăm următorul apel:

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

Arborele este prezentat în figura de mai jos:



$T(1)$, $T(2)$, și $T(3)$ fiecare constau dintr-un singur nod și deci sunt frumoși. $T(0)$ nu este frumos. Deci, procedura va returna $[0, 1, 1, 1]$.

Exemplul 2

Să considerăm următorul apel:

```
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])
```

Acest exemplu este explicat mai sus, în descrierea sarcinii.

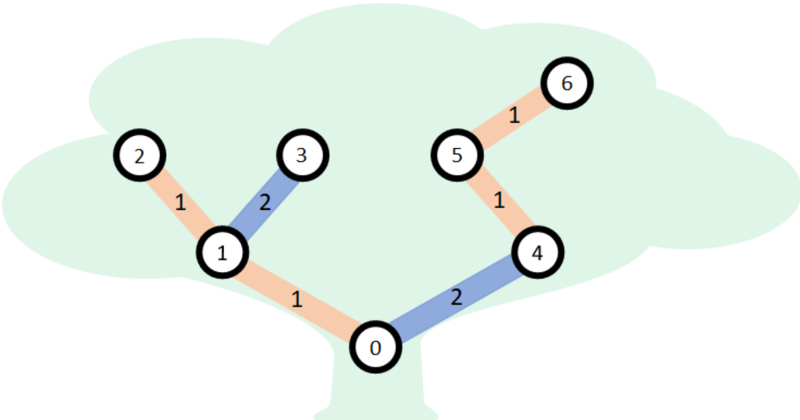
Procedura va returna $[0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$.

Exemplu 3

Să considerăm următorul apel:

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

Acest exemplu este ilustrat în următoarea figură:



$T(0)$ este unicul subarbore care este frumos. Procedura va returna $[0, 1, 1, 1, 1, 1, 1]$.

Restricții

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

Subtask-uri

1. (9 puncte) $N \leq 8$ și $M \leq 500$
2. (5 puncte) Muchia v conectează nodul v cu nodul $v - 1$. Adică, pentru fiecare v, $1 \leq v < N$, $P[v] = v - 1$.
3. (9 puncte) Fiecare nod, altul decât nodul 0 este fie conectat cu nodul 0, fie cu nodul care este conectat cu nodul 0. Adică, pentru fiecare v, $1 \leq v < N$, fie $P[v] = 0$, fie $P[P[v]] = 0$.
4. (8 puncte) Pentru fiecare c, $1 \leq c \leq M$, există cel mult două muchii de culoarea c.
5. (14 puncte) $N \leq 200$ și $M \leq 500$
6. (14 puncte) $N \leq 2\,000$ și $M = 2$
7. (12 puncte) $N \leq 2\,000$
8. (17 puncte) $M = 2$
9. (12 puncte) Fără restricții adiționale.

Exemplu de Grader

Exemplul de Grader citește inputul în următorul format:

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

Fie $b[0], \ b[1], \ \ldots$ denotă elementele tabloului returnat de `beechtree`. Exemplul de Grader tipărește răspunsul tău într-o singură linie în următorul format:

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