Beech Tree

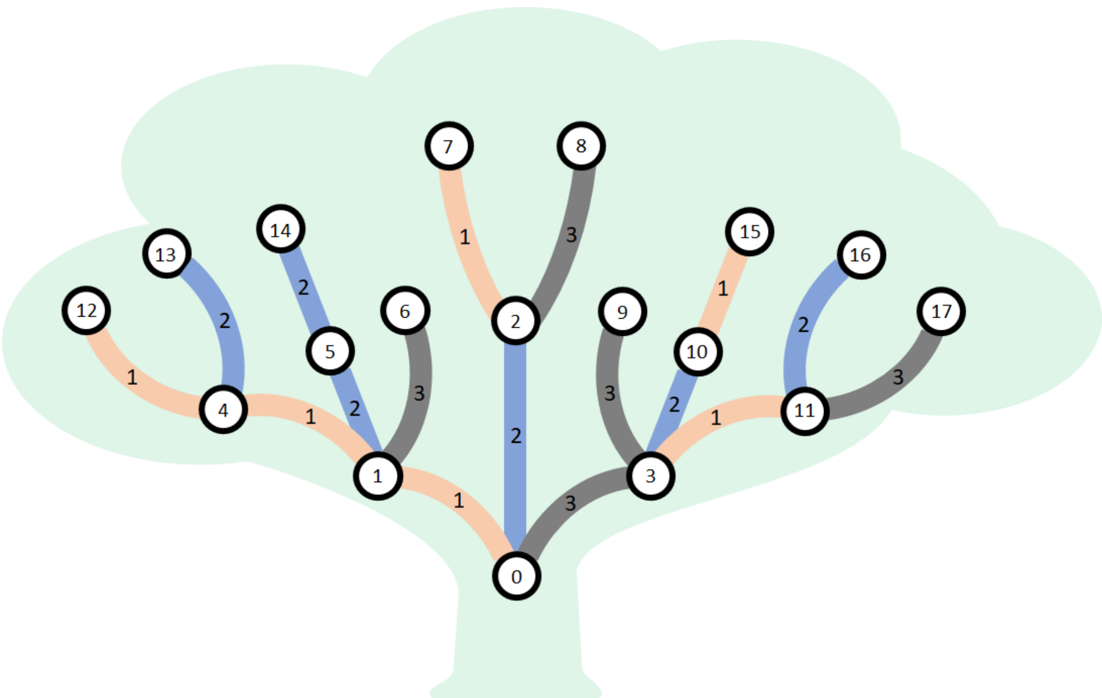
Vétyem Woods é uma famosa floresta com imensas árvores coloridas. Uma das mais altas e antigas árvores é conhecida como Ős Vezér.

A árvore Ős Vezér pode ser vista como um conjunto de N **vértices** e $N-1$ **arestas**. Os vértices são numerados de 0 a $N-1$ e as arestas numeradas de 1 a $N-1$. Cada aresta liga dois vértices distintos da árvore. Mais especificamente, a aresta i ($1 \le i < N$) liga o vértice i ao vértice $P[i]$, onde $0 \le P[i] < i$. O vértice $P[i]$ é chamado de **pai** do vértice i e o nó i é chamado de **filho** do nó $P[i]$.

Cada aresta tem uma cor. Há M possíveis cores para cada aresta, numeradas de 1 a M. A cor da aresta i é $C[i]$. Diferentes arestas podem ter a mesma cor.

Nota que segundo as definições acima, o caso $i = 0$ não corresponde a uma aresta da árvore. Por conveniência, seja $P[0] = -1$ e $C[0] = 0$.

Por exemplo, supõe que a Ős Vezér tem $N = 18$ vértices e arestas de $M = 3$ possiveis cores, com 17 arestas descritas pelas ligações $P = [-1, 0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 3, 4, 4, 5, 10, 11, 11]$ e com cores $C = [0, 1, 2, 3, 1, 2, 3, 1, 3, 3, 2, 1, 1, 2, 2, 1, 2, 3]$. A árvore pode ser visualizada na seguinte figura:



Árpád é um talentoso guarda florestal que gosta de estudar partes especificas da árvore chamadas de **subárvores**. Para cada r tal que $0 \leq r < N$, a subárvore do nó r é o conjunto $T(r)$ com as seguintes propriedades:

- O nó r pertence a $T(r)$
- Quando um nó x pertence a $T(r)$, todos os filhos de x também pertencem a $T(r)$.
- Mais nenhum nó pertence a $T(r)$.

O tamanho do conjunto $T(r)$ é denotado por $|T(r)|$.

O Árpád recentemente descobriu uma complicada mais interessante propriedade de algumas subárvores. A descoberta do Árpád envolveu brincar muito com caneta e papel, e ele suspeita que precisas de fazer o mesmo para a perceber. Ele também te vai mostrar vários exemplos que tu podes analisar em detalhe.

Supõe que fixamos um r e uma permutação $v_0, v_1, \ldots, v_{|T(r)|-1}$ dos vértices da subárvore $T(r)$.

Para cada i tal que $1 \leq i < |T(r)|$, seja $f(i)$ o número de vezes que a cor $C[v_i]$ aparece na seguinte sequência de $i-1$: cores $C[v_1], C[v_2], \ldots, C[v_{i-1}]$.

(Nota que $f(1)$ é sempre 0 porque a sequência de cores na sua definição está vazia)

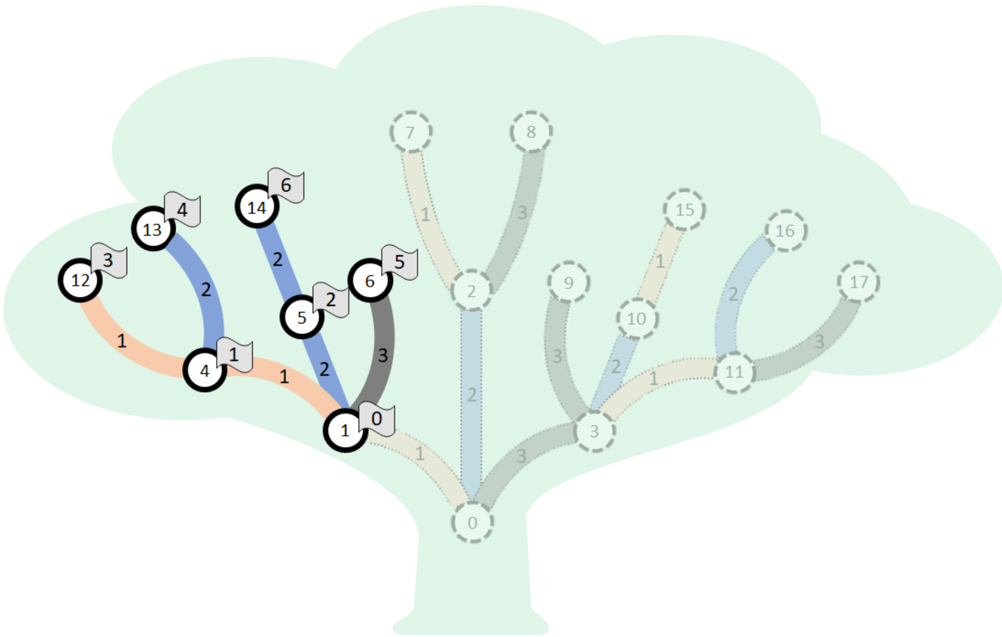
A permutação $v_0, v_1, \ldots, v_{|T(r)|-1}$ é uma **permutação bonita** se e só se todas as seguintes propriedades se verificam:

- $v_0 = r$.
- Para cada i tal que $1 \leq i < |T(r)|$, o pai do vértice v_i é o vértice $v_{f(i)}$.

Para qualquer r tal que $0 \leq r < N$, a subárvore $T(r)$ é uma **subárvore bonita** se e só se existe uma permutação bonita dos nós de $T(r)$. Nota que de acordo com a definição, qualquer subárvore que consista em um único nó é bonita.

Considera o exemplo da árvore anterior. Podemos mostrar que as subárvores $T(0)$ e $T(3)$ desta árvore não são bonitas. A subárvore $T(14)$ é bonita, visto que consiste num único vértice. De seguida, mostraremos que a subárvore $T(1)$ é também bonita.

Considera a sequência de inteiros distintos $[v_0, v_1, v_2, v_3, v_4, v_5, v_6] = [1, 4, 5, 12, 13, 6, 14]$. Esta sequência é uma permutação dos nós de $T(1)$. A figura seguinte representa esta permutação. Aa etiqueta ao lado do vértices são os índices em que esses nós aparecem na permutação.



Vamos agora verificar que esta é uma permutação bonita:

- $v_0 = r = 1$.
- $f(1) = 0$ visto que $C[v_1] = C[4] = 1$ aparece 0 vezes na sequência $[]$.
  - E de facto, temos que o pai de v_1 é v_0 (isto é, o pai de 4 é 1).
- $f(2) = 0$ visto que $C[v_2] = C[5] = 2$ aparece 0 vezes na sequência $[1]$.
  - E de facto, temos que o pai de v_2 é v_0 (isto é, o pai de 5 é 1).
- $f(3) = 1$ visto que $C[v_3] = C[12] = 1$ aparece 1 vez na sequência $[1, 2]$.
  - E de facto, temos que o pai de v_3 é v_1 (isto é, o pai de 12 é 4).
- $f(4) = 1$ visto que $C[v_4] = C[13] = 2$ aparece 1 vez na sequência $[1, 2, 1]$.
  - E de facto, temos que o pai de v_4 é v_1 (isto é, o pai de 13 é 4).
- $f(5) = 0$ visto que $C[v_5] = C[6] = 3$ aparece 0 vezes na sequência $[1, 2, 1, 2]$.
  - E de facto, temos que o pai de v_5 é v_0 (isto é, o pai de 6 é 1).
- $f(6) = 2$ visto que $C[v_6] = C[14] = 2$ aparece 2 vezes na sequência $[1, 2, 1, 2, 3]$.
  - E de facto, temos que o pai de v_6 é v_2 (isto é, o pai de 14 é 5).

Como conseguimos descobrir uma *permutação bonita* dos nós de $T(1)$, a subárvore $T(1)$ é de facto uma *subárvore bonita*.

A tua tarefa é ajudar o Árpád a decidir para cada subárvore da Ős Vezér se está é bonita ou não.

Detalhes de Implementação

Deves implementar a seguinte função.

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

- N: o número de nós da árvore.
- M: o número de possíveis cores de arestas.

- P, C: arrays de tamanho N descrevendo as arestas da árvore.
- Esta função deve devolver um array b de tamanho N. Para cada r tal que $0 \leq r < N$, $b[r]$ deve ser 1 se $T(r)$ é bonita ou 0 caso contrário.
- Esta função é chamada exatamente uma vez para cada caso de teste.

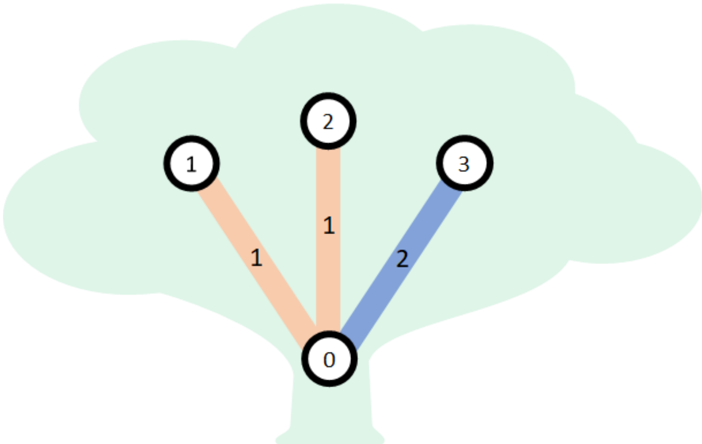
Exemplos

Exemplo 1

Considera a seguinte chamada:

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

A árvore é mostrada na figura a seguir:



$T(1)$, $T(2)$ e $T(3)$ consistem em um único nó e portanto são bonitas. $T(0)$ não é bonita. Deste modo, a função deve devolver $[0, 1, 1, 1]$.

Exemplo 2

Considera a seguinte chamada:

```
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])
```

Este exemplo foi ilustrado no enunciado.

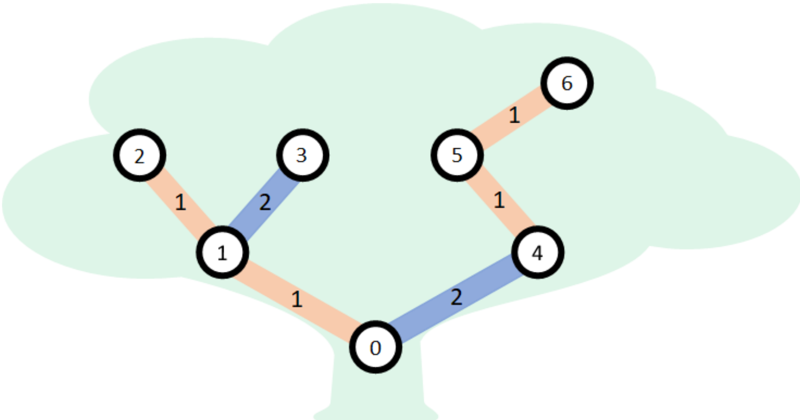
A função deve devolver $[0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$.

Exemplo 3

Considera a seguinte chamada:

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

Este exemplo é ilustrado na figura a seguir:



$T(0)$ é a única subárvore que não é bonita. A função deve devolver $[0, 1, 1, 1, 1, 1, 1]$.

Restrições

- $3 \leq N \leq 200\,000$
- $2 \leq M \leq 200\,000$
- $0 \leq P[i] < v$ (para cada i tal que $1 \leq i < N$)
- $1 \leq C[i] \leq M$ (para cada i tal que $1 \leq i < N$)
- $P[0] = -1$ e $C[0] = 0$

Subtarefas

1. (9 pontos) $N \leq 8$ e $M \leq 500$
2. (5 pontos) A aresta i liga o nó i ao nó $i - 1$. Isto é, para cada i tal que $1 \leq i < N$, $P[i] = i - 1$.
3. (9 pontos) Com a excepção do nó 0, todos os nós estão ou ligados ao nó 0, ou ligados a um nó que está ligado ao nó 0. Isto é, para cada v tal que $1 \leq v < N$, acontece que $P[v] = 0$ ou $P[P[v]] = 0$.
4. (8 pontos) Para cada c tal que $1 \leq c \leq M$, existem no máximo duas arestas com cor c.
5. (14 pontos) $N \leq 200$ e $M \leq 500$
6. (14 pontos) $N \leq 2\,000$ e $M = 2$
7. (12 pontos) $N \leq 2\,000$
8. (17 pontos) $M = 2$
9. (12 ponots) Nenhuma restrição adicional.

Avaliador Exemplo

O avaliador exemplo lê o input no seguinte formato:

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

Sejam $b[0], \ b[1], \ \ldots$ os elementos do array devolvido por `beechtree`. O avaliador exemplo escreve a tua resposta numa única linha, no seguinte formato:

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