Beech Tree

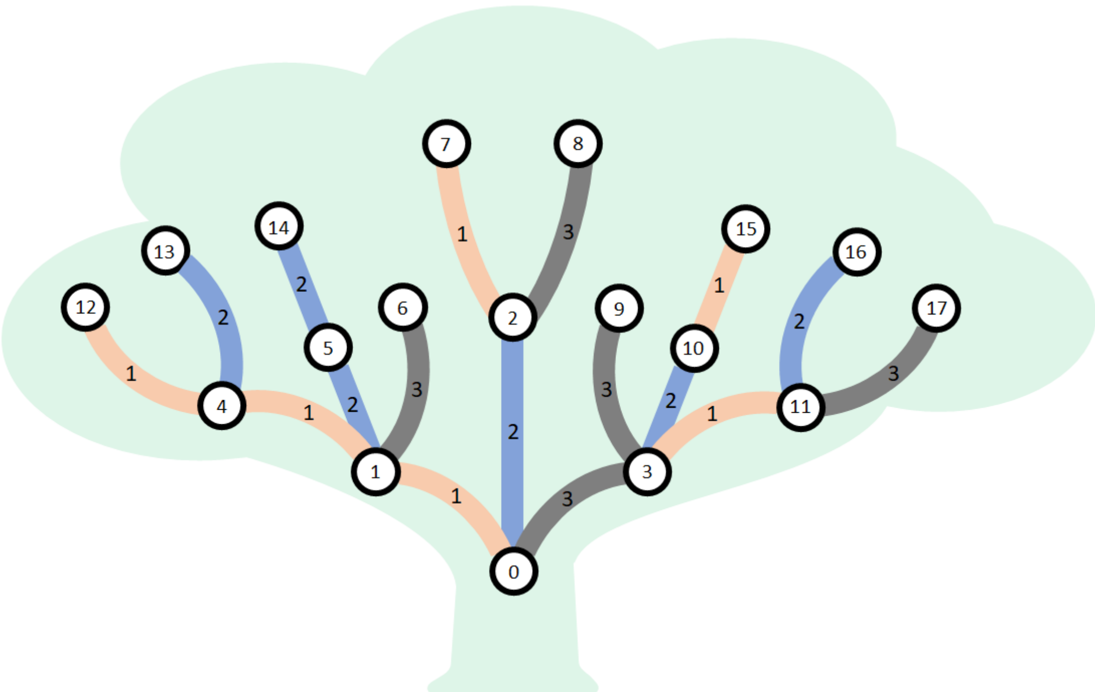
Vétyem Woods es un famoso bosque con muchos árboles coloridos. Una de las hayas (un tipo de árbol) más antiguas y altas se llama Ős Vezér.

El árbol Ős Vezér puede ser modelado como un conjunto de N **nodos** y $N-1$ **aristas**. Los nodos están enumerados desde 0 hasta $N-1$ y las aristas están enumeradas desde 1 hasta $N-1$. Cada arista conecta dos nodos distintos del árbol. Específicamente, la arista i ($1 \leq i < N$) conecta el nodo i al nodo $P[i]$, donde $0 \leq P[i] < i$. El nodo $P[i]$ es llamado **padre** del nodo i, y el nodo i es llamado **hijo** del nodo $P[i]$.

Cada arista tiene un color. Hay M posibles colores de aristas enumerados desde el 1 hasta M. El color de la arista i es $C[i]$. Aristas diferentes pueden tener el mismo color.

Observa que en la definición de arriba, el caso $i = 0$ no corresponde a una arista del árbol. Por conveniencia, nosotros dejamos $P[0] = -1$ y $C[0] = 0$.

Por ejemplo: supongamos que Ős Vezér tiene $N = 18$ nodos y $M = 3$ posibles colores de aristas, con 17 aristas con su conexión descrita por $P = [-1, 0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 3, 4, 4, 5, 10, 11, 11]$ y sus colores $C = [0, 1, 2, 3, 1, 2, 3, 1, 3, 3, 2, 1, 1, 2, 2, 1, 2, 3]$. El árbol es mostrado en la siguiente imagen:



Árpád es un talentoso guardabosques que le gusta estudiar la parte específica de los árboles llamada **sub-árbol**. Por cada r tal que $0 \leq r < N$, el sub-árbol del nodo r es el conjunto $T(r)$ de nodos con las siguientes propriedades:

- El nodo r pertenece a $T(r)$.
- Cuando un nodo x pertenece a $T(r)$, todos los hijos de x también pertenecen a $T(r)$.
- Ningún otro nodo pertenece a $T(r)$.

El tamaño del conjunto $T(r)$ es denotado como $|T(r)|$.

Árpád ha descubierto una complicada pero interesante propriedad de los sub-árboles. Su descubrimiento involucró mucho uso de lápiz y papel, y él sospecha que podrías necesitar hacer lo mismo para entenderlo. Él también te mostrará múltiples ejemplos que podrás analizar a detalle.

Supóngase que se tiene un r fijo y una permutación $v_0, v_1, \ldots, v_{|T(r)|-1}$ de los nodos en el sub-árbol $T(r)$.

Para cada i tal que $1 \leq i < |T(r)|$, sea $f(i)$ el número de veces que el color $C[v_i]$ aparece en la siguiente secuencia de $i-1$ colores: $C[v_1], C[v_2], \ldots, C[v_{i-1}]$.

(Nota que $f(1)$ siempre es 0 porque la secuencia de colores en su definición es vacía.)

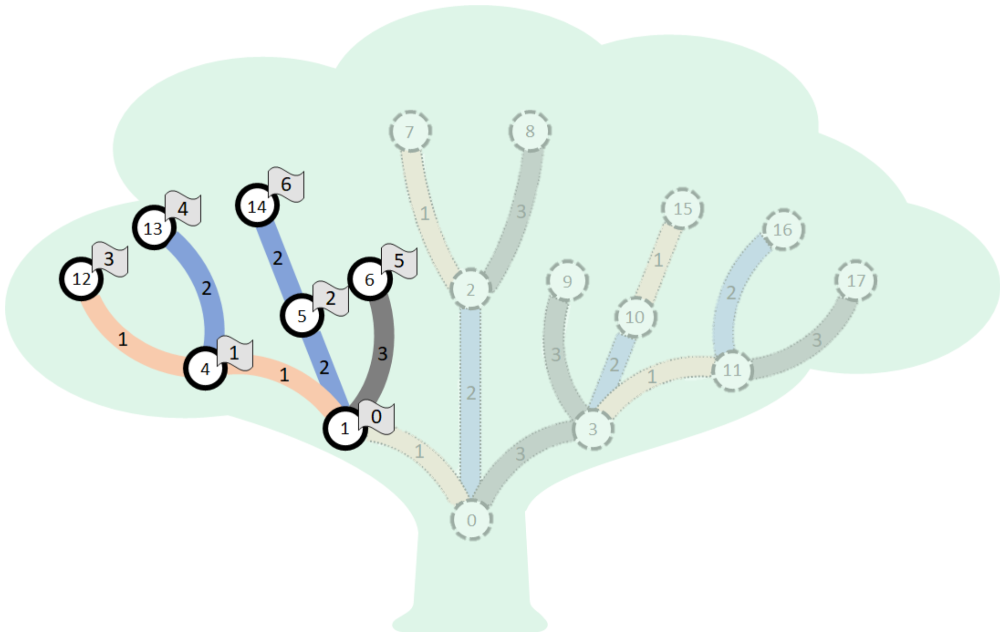
La permutación $v_0, v_1, \ldots, v_{|T(r)|-1}$ es una **permutación hermosa** si y solo si todas las siguientes propriedades se cumplen:

- $v_0 = r$.
- Para cada i tal que $1 \leq i < |T(r)|$, el padre del nodo v_i es el nodo $v_{f(i)}$.

Para cualquier r tal que $0 \leq r < N$, el sub-árbol $T(r)$ es un **sub-árbol hermoso** si y solo si existe una permutación hermosa de los nodos en $T(r)$. Nótese que de acuerdo a la definición, cada sub-árbol que consiste de un solo nodo es hermoso.

Considera el ejemplo del árbol de más arriba. Puede ser demostrado que los sub-árboles $T(0)$ y $T(3)$ de este árbol no son hermosos. El sub-árbol $T(14)$ es hermoso, dado que consiste de un solo nodo. Abajo mostraremos que el sub-árbol $T(1)$ es también hermoso.

Considera la secuencia de distintos enteros $[v_0, v_1, v_2, v_3, v_4, v_5, v_6] = [1, 4, 5, 12, 13, 6, 14]$. Esta secuencia es una permutación de los nodos en $T(1)$. La imagen de abajo muestra esta permutación. Las etiquetas adjuntas a los nodos son los índices en los cuales estos nodos aparecen en la permutación.



Ahora verificaremos que ésta es una *permutación hermosa*.

- $v_0 = 1$.
- $f(1) = 0$ dado que $C[v_1] = C[4] = 1$ aparece 0 veces en la secuencia $[]$.
    - Por lo tanto, el padre de v_1 es v_0. Es decir, el padre del nodo 4 es el nodo 1. (Formalmente, $P[4] = 1$.)
- $f(2) = 0$ dado que $C[v_2] = C[5] = 2$ aparece 0 veces en la secuencia $[1]$.
    - Por lo tanto, el padre de v_2 es v_0. Es decir, el padre de 5 es 1.
- $f(3) = 1$ dado que $C[v_3] = C[12] = 1$ aparece 1 vez en la secuencia $[1, 2]$
    - Por lo tanto, el padre de v_3 es v_1. Es decir, el padre de 12 es 4.
- $f(4) = 1$ dado que $C[v_4] = C[13] = 2$ aparece 1 vez en la secuencia $[1, 2, 1]$
    - Por lo tanto, el padre de v_4 es v_1. Es decir, el padre de 13 es 4.
- $f(5) = 0$ dado que $C[v_5] = C[6] = 3$ aparece 0 veces en la secuencia $[1, 2, 1, 2]$
    - Por lo tanto, el padre de v_5 es v_0. Es decir, el padre de 6 es 1.
- $f(6) = 2$ dado que $C[v_6] = C[14] = 2$ aparece 2 veces en la secuencia $[1, 2, 1, 2, 3]$
    - Por lo tanto, el padre de v_6 es v_2. Es decir, el padre de 14 es 5.

Como pudimos encontrar una *permutación hermosa* de los nodos en $T(1)$, el sub-árbol $T(1)$ es un *sub-árbol hermoso*.

Tu tares es ayudar a que Árpád decida por cada sub-árbol de Ős Vezér si es hermoso.

Detalles de Implementación

Tu debes implementar el siguiente procedimiento:

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

- N: el número de nodos en el árbol.

- M: el número de posibles colores de las aristas.
- P, C: arreglos de tamaño N describiendo las aristas del árbol.
- Este procedimiento deberá retornar un arreglo b de tamaño N. Por cada r tal que $0 \leq r < N$, $b[r]$ debiera ser 1 si $T(r)$ es hermoso, y 0 en caso contrario.
- Este procedimiento es llamado exactamente una vez por cada caso de prueba.

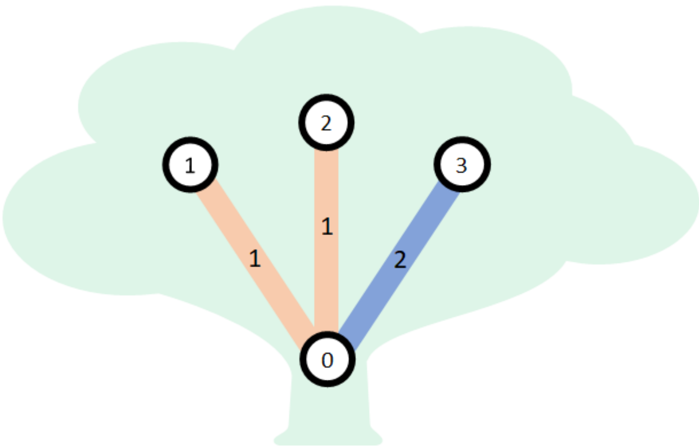
Ejemplos

Ejemplos 1

Considera la siguiente llamada:

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

El árbol se muestra en la siguiente imagen:



$T(1)$, $T(2)$, y $T(3)$ cada uno contiene un sólo nodo y por lo tanto son cada uno hermosos. $T(0)$ no es hermoso. Por lo tanto, el procedimiento debiera retornar $[0, 1, 1, 1]$.

Ejemplo 2

Considera la siguiente llamada:

```
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 ejemplo está ilustrado en la tarea descrita arriba.

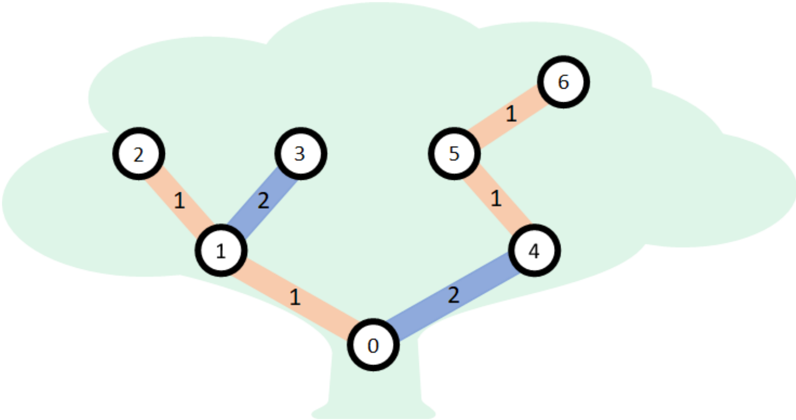
El procedimiento debiera retornar $[0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$.

Ejemplo 3

Considere la siguiente llamada:

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

Este ejemplo está ilustrado en la siguiente imagen:



$T(0)$ es el único sub-árbol que no es hermoso. El procedimiento debiera retornar $[0, 1, 1, 1, 1, 1, 1]$.

Restricciones

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

Sub-tareas

1. (9 puntos) $N \le 8$ y $M \le 500$
2. (5 puntos) La arista i conecta el nodo i con el nodo $i-1$. Esto es, por cada i tal que $1 \le i < N$, $P[i] = i - 1$.
3. (9 puntos) Cada nodo que no sea el nodo 0 está conectado al nodo 0 o está conectado a un nodo que está conectado al nodo 0. Esto es, por cada i tal que $1 \le i < N$, para cualquier $P[i] = 0$ o $P[P[i]] = 0$.
4. (8 puntos) Para cada c tal que $1 \le c \le M$, hay como máximo dos aristas de color c.
5. (14 puntos) $N \le 200$ y $M \le 500$
6. (14 puntos) $N \le 2\,000$ y $M = 2$
7. (12 puntos) $N \le 2\,000$
8. (17 puntos) $M = 2$
9. (12 puntos) Sin restricciones adicionales.

Evaluador de ejemplo

El evaluador de ejemplo lee la entrada en el siguiente formato:

- linea 1: $N \; M$

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

Vea que $b[0],\ b[1],\ \ldots$ denota los elementos del arreglo que retorna `beechtree`. El evaluador de ejemplo imprime tus respuestas en una sola linea, en el siguiente formato:

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