Árbol de Hayas

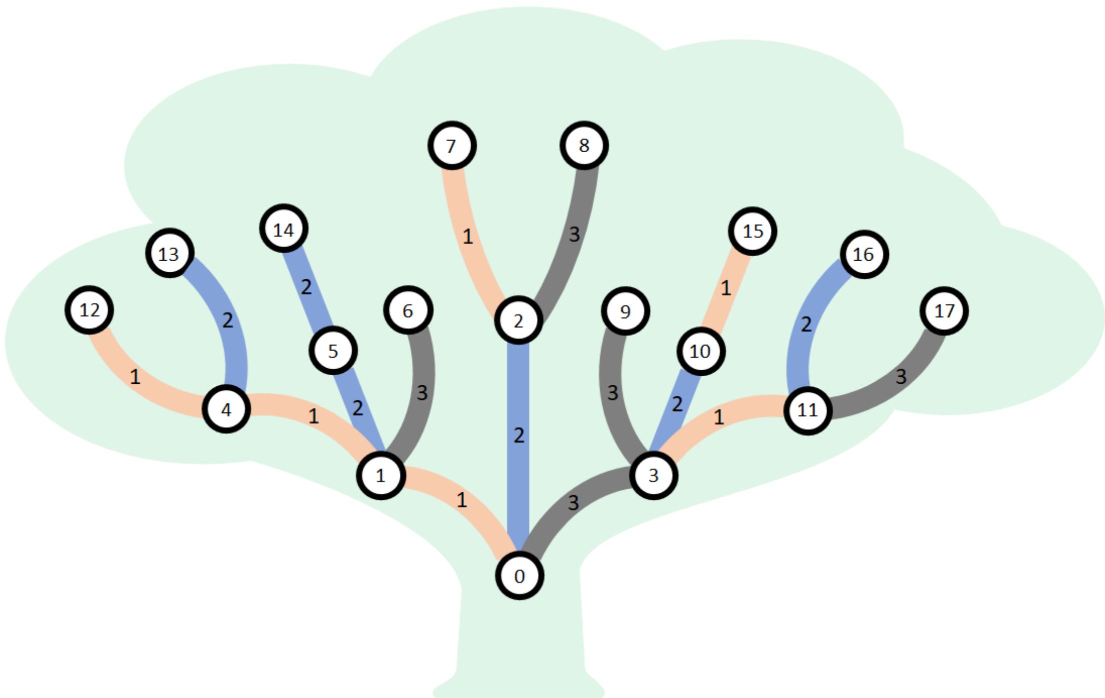
Vétyem Woods es un bosque famoso con muchos árboles coloridos. Una de las hayas (árbol europeo) más antiguas y altas se llama Ős Vezér.

El árbol Ős Vezér puede modelarse como un conjunto de N **nodos** y $N-1$ **aristas**. Los nodos están numerados del 0 al $N-1$ y las aristas están numeradas del 1 al $N-1$. Cada arista conecta a dos nodos distintos del árbol. Específicamente la arista i ($1 \leq i < N$) conecta el nodo i con el nodo $P[i]$, donde $0 \leq P[i] < i$. Al nodo $P[i]$ se le conoce como el **padre** del nodo i, y al nodo i se le conoce como el **hijo** del nodo $P[i]$.

Cada arista tiene un color. Hay M posibles colores de arista numerados del 1 al M. El color de la arista i es $C[i]$. Puede ser que distintas aristas tengan el mismo color.

Ten en cuenta que en las definiciones anteriores, el caso $i = 0$ no corresponde con ninguna arista del árbol. Por conveniencia, definimos $P[0] = -1$ y $C[0] = 0$.

Por ejemplo, supongamos que Ős Vezér tiene $N = 18$ nodos y $M = 3$ colores de aristas posibles, con 17 aristas descritas por las conexiones $P = [-1, 0, 0, 0, 1, 1, 1, 2, 2, 3, 3, 3, 4, 4, 5, 10, 11, 11]$ y colores $C = [0, 1, 2, 3, 1, 2, 3, 1, 3, 3, 2, 1, 1, 2, 2, 1, 2, 3]$. El árbol resultante se muestra en la siguiente figura:



Árpád es un ingeniero forestal talentoso a quien le gusta estudiar partes específicas del árbol llamadas **subárboles**. Para toda r tal que $0 \leq r < N$, el subárbol del nodo r es el conjunto $T(r)$ de nodos que cumplen las siguientes propiedades:

- El nodo r pertenece a $T(r)$
- Si el nodo x peretence a $T(r)$, todos los hijos de x también pertenecen a $T(r)$.
- Ningún otro nodo pertenence a $T(r)$.

El tamaño del conjunto $T(r)$ se denota como $|T(r)|$.

Recientemente, Árpád descubrió una propiedad complicada pero interesante de los subárboles. Su descubrimiento requirió mucha prueba y error a lápiz y papel, y sospecha que tú también necesitarás hacer lo mismo. También te mostrará multiples ejemplos que después puedes analizar con más detalle.

Supón que tenemos un r fijo y una permutación $v_0, v_1, \ldots, v_{|T(r)|-1}$ de los nodos en el subárbol $T(r)$.

Para toda 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 la definción anterior es vacía.)

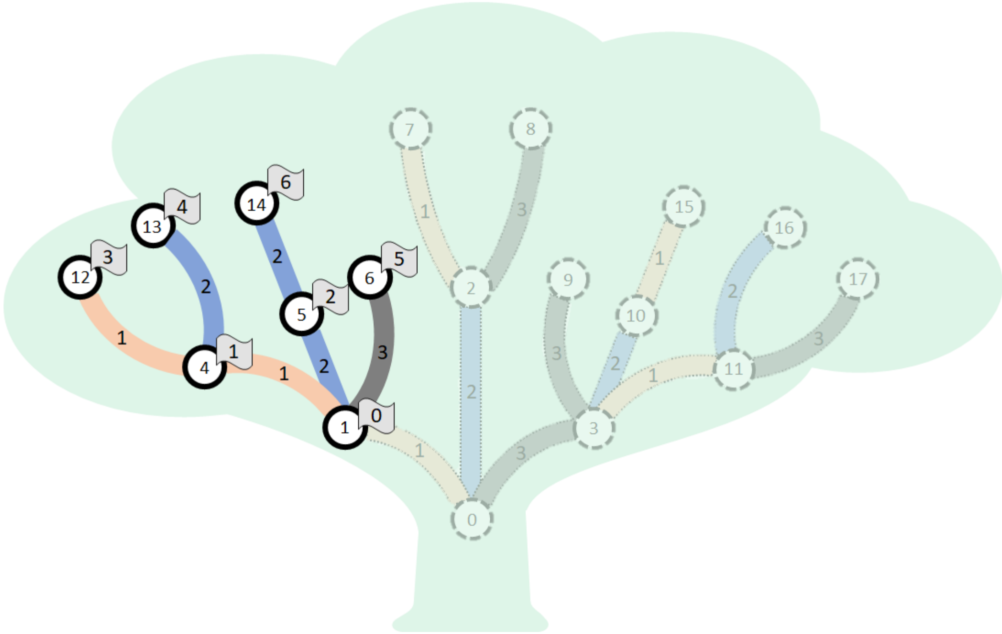
La permutación $v_0, v_1, \ldots, v_{|T(r)|-1}$ es una **permutación bonita** si y solo si se cumplen todas las siguientes propiedades:

- $v_0 = r$.
- Para toda 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 subárbol $T(r)$ es un **subárbol bonito** si y solo si existe una permutación bonita de los nodos en $T(r)$. Nota que, de acuerdo a la definición, todos los subárboles que consisten de un solo nodo son bonitos.

Considera el árbol de ejemplo anterior. Es posible demostrar que los subárboles $T(0)$ y $T(3)$ de este árbol no son bonitos. El subárbol $T(14)$ es bonito, pues consiste de un solo nodo. A continuación, mostraremos que el subárbol $T(1)$ también es bonito.

Considera la secuencia de enteros distintos $[v_0, v_1, v_2, v_3, v_4, v_5, v_6] = [1, 4, 5, 12, 13, 6, 14]$. Esta secuencia es una permutación de los nodos en $T(1)$. La siguiente figura muestra esta permutación. Las etiquetas al lado de cada nodo representan los índices de estos nodos en la permutación.



Ahora verificaremos que es una **permutación bonita**.

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

Dado que pudimos encontrar una **permutación bonita** de los nodos en $T(1)$, el subárbol $T(1)$ es un **subárbol bonito**.

Tu tarea es ayudar a Árpád a decidir, para cada subárbol de Ős Vezér, si es bonito o no.

Detalles de implementación

Debes implementar la siguiente función:

```
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 arista.
- P, C: arreglos de tamaño N describiendo las aristas del árbol.
- Esta función deberá devolver un arreglo b de tamaño N. Para toda r tal que $0 \leq r < N$, $b[r]$ debe ser 1 si $T(r)$ es bonito y 0 de lo contrario.
- Esta función se llamará exactamente una vez para cada caso de prueba.

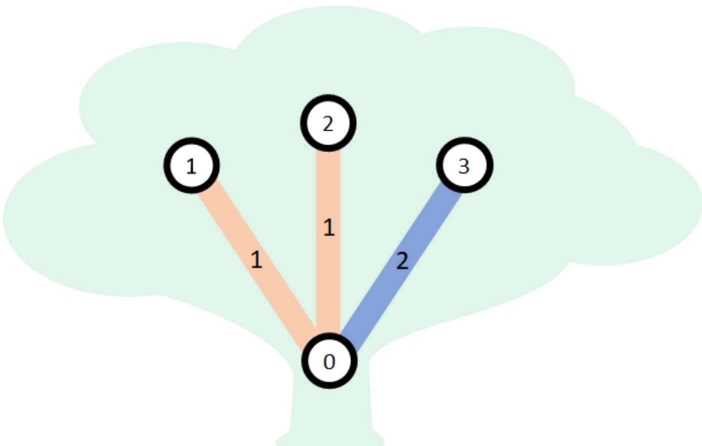
Ejemplos

Ejemplo 1

Considera la siguiente llamada a la función:

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

El árbol se muestra en la siguiente figura:



$T(1)$, $T(2)$ y $T(3)$ consisten de un solo nodo cada uno y por lo tanto son bonitos. $T(0)$ no es bonito. Por lo tanto, la función deberá retornar $[0, 1, 1, 1]$.

Ejemplo 2

Considera la siguiente llamada a la función:

```
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 se ilustra en la descripción del problema.

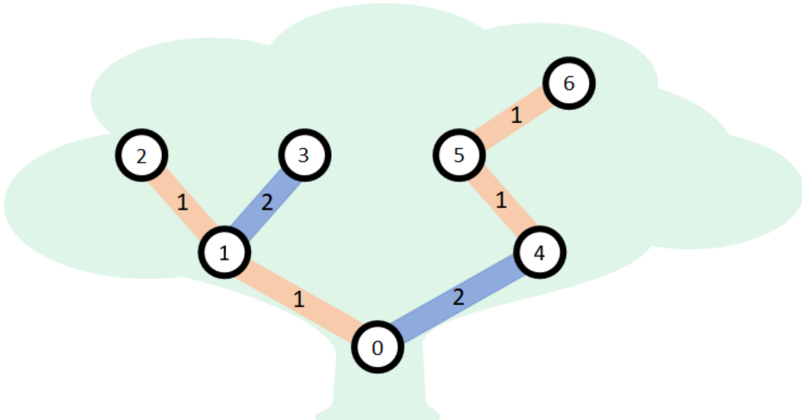
La función deberá retornar $[0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$.

Ejemplo 3

Considera la siguiente llamada a la función:

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

Este ejemplo se ilustra en la siguiente figura.



$T(0)$ es el único subárbol que no es bonito. La función deberá retornar $[0, 1, 1, 1, 1, 1, 1]$.

Límites

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

Subtareas

1. (9 puntos) $N \leq 8$ y $M \leq 500$
2. (5 puntos) La arista i conecta el nodo i con el nodo $i - 1$. Es decir, para toda i tal que $1 \leq i < N$, $P[i] = i - 1$.
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. Es decir que para toda i tal que $1 \leq i < N$, se cumple que o $P[i] = 0$ o $P[P[i]] = 0$.
4. (8 puntos) Para toda c tal que $1 \leq c \leq M$, hay a lo más dos aristas de color c.
5. (14 puntos) $N \leq 200$ y $M \leq 500$
6. (14 puntos) $N \leq 2\,000$ y $M = 2$
7. (12 puntos) $N \leq 2\,000$
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:

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

Sean $b[0],\ b[1],\ \ldots$ los elementos del arreglo devuelto por `beechtree`. El evaluador de ejemplo imprime tu respuesta en una sola línea, en el siguiente formato:

- línea 1: $b[0]\ b[1]\ \ldots$