

Búsqueda del tesoro

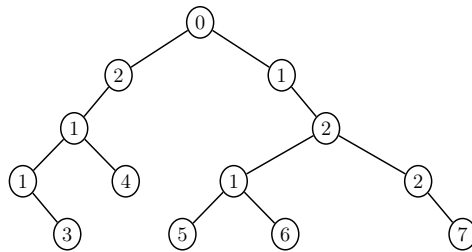
Dado un árbol binario, en cuya raíz se encuentra situado un tesoro y cuyos nodos internos pueden contener un dragón o no contener nada, se pide diseñar un algoritmo que nos indique la hoja del árbol cuyo camino hasta la raíz tenga el menor número de dragones. En caso de que existan varios caminos con el mismo número de dragones, el algoritmo devolverá el que se encuentre más a la izquierda de todos ellos.

Para ello implementar una función que reciba un árbol binario cuyos nodos almacenan enteros:

1. La raíz contiene el entero 0, que representa al tesoro.
2. Los nodos internos contienen el entero 1 para indicar que en el nodo hay un dragón o el entero 2 para indicar que no hay dragón.
3. En cada hoja se almacena un entero mayor o igual a 3 que no puede estar repetido.

y devuelva el entero de la hoja del camino seleccionado. El árbol tiene como mínimo un nodo raíz y un nodo hoja diferente de la raíz.

Por ejemplo, dado el siguiente árbol (segundo caso de prueba mostrado en el ejemplo) el algoritmo devolverá el entero 4:



Entrada

La primera línea contiene el número de casos de prueba. Las líneas siguientes contienen cada una un caso de prueba. Cada una comienza con el contenido de la raíz del árbol, al que le sigue la descripción del hijo izquierdo y después la del hijo derecho. El número -1 indica la ausencia de hijo o lo que es lo mismo, el árbol vacío.

Salida

Una línea para cada caso de prueba con el número de la hoja resultado.

Entrada de ejemplo

```
2
0 3 -1 -1 4 -1 -1
0 2 1 1 -1 3 -1 -1 4 -1 -1 -1 1 -1 2 1 5 -1 -1 6 -1 -1 2 -1 7 -1 -1
```

Salida de ejemplo

```
3
4
```

Nota

Este ejercicio debe verse en el contexto de la asignatura de Estructura de Datos y Algoritmos (EDA), FDI-UCM 2016/2017 (prof. Clara Maria Segura Diaz). Por tanto *no* vale cualquier solución, sino sólo aquellas que utilicen los conceptos de EDA. Es muy posible que se den aclaraciones adicionales en clase a este respecto.