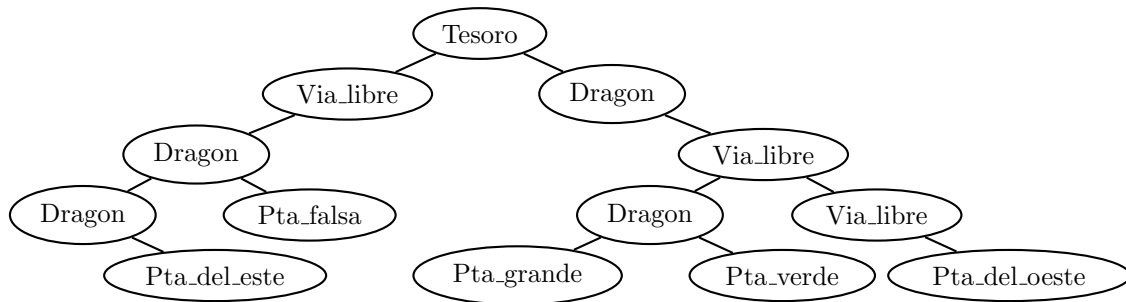


Laberintos con dragones

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, el algoritmo devolverá el que se encuentra más a la izquierda de todos. (Examen de EDA. Junio 2014)



Requisitos de implementación.

Se extenderá la clase `binTree` vista en clase con un nuevo método, `dragones` que resuelva el problema. Se indicará en un comentario el coste de la función implementada, para árboles equilibrados y para el caso peor.

Entrada

La entrada comienza con el número de casos de prueba. Para cada caso se muestra el recorrido en preorden del árbol (el recorrido puede mostrarse en varias líneas). El nodo raíz se representa con la cadena `tesoro`; los nodos internos pueden ser `Dragon` o `Via_libre`; los nodos hoja pueden tener cualquier identificador diferente de los anteriores.

El árbol tiene como mínimo un nodo raíz y un nodo hoja diferente de la raíz. Los punteros a nulo se representan con tres puntos seguidos.

Salida

Para cada caso de prueba se escribirá el valor de la hoja que identifica la rama de coste mínimo y el número de dragones en dicha rama.

Entrada de ejemplo

```
1
Tesoro Via_libre Dragon Dragon ... Puerta_del_este ... ... Puerta_falsa ... ...
Dragon ... Via_libre Dragon Puerta_grande ... ... Puerta_verde ... ... Via_libre
... Puerta_del_oeste ... ...
```

Salida de ejemplo

```
Puerta_falsa 1
```

Autor: Isabel Pita