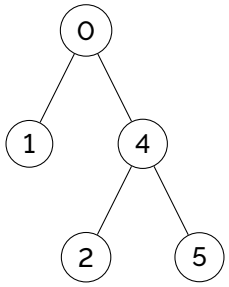
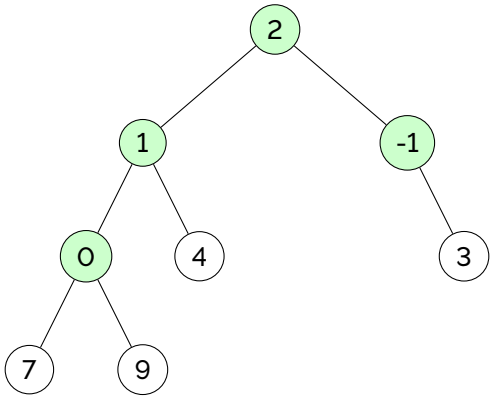
Factor de equilibrio en un árbol binario

En un árbol binario, el *factor de equilibrio en número de hojas* de un nodo interno es la diferencia entre el número de hojas del subárbol izquierdo y el del subárbol derecho. Por ejemplo, dado el siguiente árbol:



El factor de equilibrio de la raíz es `-1`, que es la diferencia entre el número de hojas del subárbol izquierdo (tiene 1 hoja) y el del árbol derecho (que tiene 2 hojas). El factor de equilibrio del nodo `4` es `0`, ya que ambos hijos tienen el mismo número de hojas.

Los *árboles de equilibrio* (¡no confundir con árboles equilibrados!) son aquellos en los que los valores de sus nodos internos son, precisamente, sus respectivos factores de equilibrio. Por ejemplo, el siguiente árbol es un árbol de equilibrio:



Los nodos resaltados son los nodos internos del árbol, que son los que han de tener un valor igual al factor de equilibrio correspondiente. Los valores de las hojas son irrelevantes a la hora de determinar si un árbol es de equilibrio o no.

En este ejercicio se pide:

1. Definir una función `arbol_de_equilibrio` con la siguiente cabecera:

   ```
   bool arbol_de_equilibrio(const BinTree<int> &tree);
   ```

   Esta función debe devolver `true` si el árbol pasado como parámetro es un árbol de equilibrio, o `false` en caso contrario.

2. Indicar el coste, en el caso peor, de la función anterior. El coste debe estar expresado en función del número de nodos del árbol de entrada.

Entrada

La entrada comienza con un número que indica el número de casos de prueba que vienen a continuación. Cada caso de prueba consiste en una línea con la descripción de un árbol binario mediante la notación vista en clase. El árbol vacío se representa mediante `.` y el árbol no vacío mediante `(iz x dr)`, siendo `x` la raíz, y `iz`, `dr` las representaciones de ambos hijos.

Salida

Para cada árbol se escribirá una línea con la cadena `SI` si es un árbol de equilibrio, o `NO` en caso contrario.

Entrada de ejemplo

Entrada de ejemplo

```
4
((((. 7 .) 0 (. 9 .)) 1 (. 4 .)) 2 (. -1 (. 3 .)))
(((. 14 .) 0 (. 19 .)) 0 (. 21 .))
(. 27 .)
.
```

Salida de ejemplo

```
SI
NO
SI
SI
```