

# Árboles genealógicos

Dado un árbol binario de enteros, se entiende que es un *árbol genealógico correcto* si cumple las siguientes reglas, producto de interpretar el entero de cada nodo como la edad del individuo, el hijo izquierdo como su primer hijo de un máximo de dos, y el hijo derecho como el segundo hijo:

1. La edad del padre siempre supera en al menos 18 años las edades de cada uno de los hijos (si existen).
2. La edad del segundo hijo (si existe) es al menos dos años menos que la del primer hijo (no hay hermanos gemelos/mellizos en estos árboles).
3. Los árboles genealógicos de ambos hijos son también correctos.

Dado un árbol binario queremos averiguar si es o no un árbol genealógico correcto, y en caso de serlo, cuál es el número de generaciones distintas que hay en la familia (niveles del árbol). El árbol vacío se considera árbol genealógico correcto sin generaciones.

*Requisitos de implementación.*

Se implementará una función *externa* a la clase `bintree` que explore el árbol averiguando si es genealógico o no, y cuántas generaciones contiene. Esta función debe tener un coste lineal con respecto al número de nodos del árbol.

## Entrada

La primera línea contiene el número de casos de prueba. Las líneas siguientes contienen cada una un caso de prueba (un árbol binario). Cada una comienza con el contenido de la raíz del árbol (un entero positivo), 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 la cadena `SI` en caso de ser el árbol genealógico, seguida del número de generaciones; y `NO` en caso de no serlo.

## Entrada de ejemplo

```
5
40 20 2 -1 -1 0 -1 -1 -1
23 3 -1 -1 2 -1 -1
90 60 35 -1 -1 30 11 -1 -1 8 -1 -1 57 39 15 -1 -1 -1 28 10 -1 -1 6 -1 -1
20 -1 1 -1 -1
-1
```

## Salida de ejemplo

```
SI 3
NO
SI 4
NO
SI 0
```

**Autores:** Clara Segura y Alberto Verdejo.