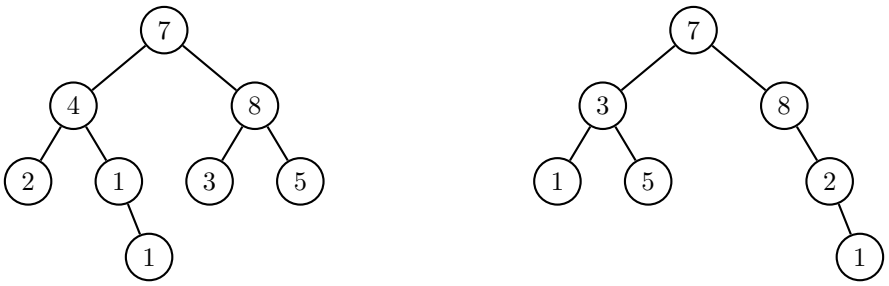
Problema B

Dado un árbol binario de enteros, se dice que es *diestro* si o bien es el árbol vacío o una hoja, o bien la suma de los valores en todos los nodos del hijo derecho es mayor que la suma de los valores de todos los nodos de su hijo izquierdo y, además, tanto el hijo izquierdo como el derecho son diestros.

Por ejemplo, de los siguienes árboles, el de la izquierda no es diestro porque el subárbol con raíz 4 no lo es (los descendientes en sus dos hijos suman lo mismo). El de la derecha sí es diestro (todos los nodos cumplen que la suma de los descendientes en su hijo derecho es mayor que la suma de los descendientes en su hijo izquierdo).



Dado un árbol binario queremos averiguar si es diestro o no.

Requisitos de implementación.

Se debe implementar una función *externa* a la clase `bintree` que explore el árbol de manera eficiente averiguando si es diestro o no. La función no podrá tener parámetros de entrada/salida.

Entrada

La entrada comienza con el número de casos que vienen a continuación. Cada caso de prueba consiste en una línea con la descripción de un árbol binario: si el árbol es vacío se representa con un –1; si no, primero aparece su raíz, y a continuación la descripción del hijo izquierdo y después la del hijo derecho, dadas de la misma manera.

Salida

Para cada árbol, se escribirá una línea con un `SI` si el árbol es diestro y un `NO` si no lo es.

Entrada de ejemplo

```
4
7 4 2 -1 -1 1 -1 1 -1 -1 8 3 -1 -1 5 -1 -1
7 3 1 -1 -1 5 -1 -1 8 -1 2 -1 1 -1 -1
-1
5 2 -1 -1 2 -1 -1
```

Salida de ejemplo

```
NO
SI
SI
NO
```