

---

## Determinar si un arbre general és binari

T10651\_ca

---

Afegeix un **mètode públic** a la classe `ArbreGen<T>` que determini si l'arbre és binari. Un node d'un arbre general representa un arbre binari si és buit, o bé no té fills (és una fulla), o bé en té exactament 2.

La capçalera és la següent:

```
/**
 * @brief Determina si un arbre general és binari. Un node d'un arbre
 * general representa un arbre binari quan és buit, o bé no té fills,
 * o bé en té exactament 2.
 *
 * @returns 'true' si l'arbre general és binari, 'false' altrament.
 */
bool is_binary() const;
```

### Observació

Malgrat els arbres binaris poden tenir només un fill (quan alguna de les seves dues branques és un arbre buit) en la definició de més amunt considerarem que els arbres amb un sol fill no són binaris perquè en arbres generals no podem distingir entre fill dret o esquerre.

Per poder avaluar l'ús de punters, *no feu servir altres mètodes, ni públics ni privats*, de la classe per resoldre el problema, accediu sempre als membres privats.

Recordeu que un arbre general no té fills buits.

Els fitxers públics (icona del gatet) contenen:

<code>ArbreG.hh</code>	la classe <code>ArbreGen&lt;T&gt;</code>
<code>ArbreG-io.hh</code>	entrada/sortida per <code>ArbreGen&lt;T&gt;</code>
<code>main.cc</code>	el programa principal (gestiona l'entrada i sortida)
<code>Makefile</code>	per compilar amb <code>make</code> al terminal
<code>.vscode</code>	per compilar i debuggar amb F5

Per entregar només cal **enviar el fitxer** `ArbreG.hh` **modificat**.

### Entrada

De l'entrada se n'encarrega ja el programa principal. L'entrada comença amb un nombre  $n$  que indica quants casos té segueixen. Després hi ha els  $n$  arbres generals en el format gràfic típic de la classe `Tree<T>` de PRO2. Els diferents cassos estan separats per una línia buida.

### Sortida

De la sortida també se n'encarrega el programa principal. La sortida per a cada arbre és "si" o bé "no", en funció del resultat de cridar al mètode `is_binary` amb l'arbre llegit a l'entrada.

### Exemple d'entrada

```
5
3
'-- 5

1
|-- 2
'-- 3

4
|-- 5
|-- 1
'-- 7
    |-- 2
    |-- 9
    |   '-- 3
    |-- 2
    '-- 4

1
|-- 2
|   |-- 3
|   '-- 4
'-- 4
    |-- 5
    '-- 6

8
|-- 3
|-- 1
'-- 6
    |-- 8
    |-- 2
    |   '-- 4
    |-- 1
    '-- 5
```

### Exemple de sortida

```
no
si
no
si
no
```

### Informació del problema

Autor : M<sup>a</sup> Lluïsa Bonet i Pau Fernández  
Generació : 2025-06-11 10:07:40

© *Jutge.org*, 2006–2025.  
<https://jutge.org>