

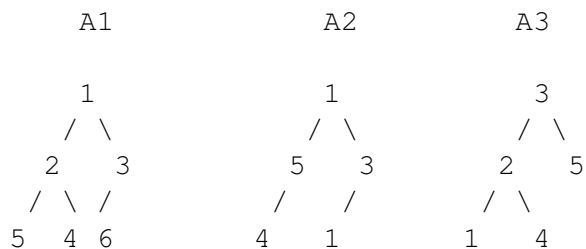
Arbre Binari Complet

X78267_ca

Un arbre binari és un arbre d'aritat 2. Una **fulla** d'un arbre és un node que no té cap fill. La **profunditat** d'una fulla és el nombre d'arestes que hi ha entre l'arrel de l'arbre i la fulla. Un arbre binari **complet** és un arbre binari que té les següents propietats:

1. La diferència entre la fulla més profunda i la fulla menys profunda és, com a màxim, 1.
2. Les fulles de l'últim nivell són totes el més a l'esquerra possible.

Per exemple,



tenim que l'arbre A1 és un arbre complet, ja que compleix la condició (1) ja que totes les fulles tenen la mateixa profunditat, i també la (2) ja que totes les fulles del nivell més profund són totes a l'esquerra.

En canvi, l'arbre A2 no és complet perquè si bé compleix la condició (1) perquè totes dues fulles (4 i 1) tenen la mateixa profunditat, no compleix la condició (2), ja que el node 1 no és tan a l'esquerra com podria (hauria de ser el fill dret de 5).

L'arbre A3 és complet perquè compleix la condició (1) ja que diferència entre la profunditat de la fulla 5 (la menys profunda) i la de les fulles 1 o 4 és 1, i també compleix la condició 2, ja que les fulles del nivell més profund són totes dues a l'esquerra.

Feu la funció

```
bool arbreBinariComplet (arbreBin<int>);
```

tal que, donat un arbre binari d'enters, torni un booleà que sigui cert si i només si l'arbre és un arbre binari complet.

Per a A1 tornaria `true`, per a A2 tornaria `false` i per a A3 tornaria `true`.

Entrada

La funció rep un arbres binari d'enters A.

Sortida

`true` si i només si A és un arbre binari complet.

Observació

Heu d'enviar la solució comprimida en un fitxer `.tar`:

```
tar cvf program.tar arbreBinariComplet.cpp
```

Observeu que per compilar us donem el `Makefile`,

la capçalera del mòdul funcional `arbreBinariComplet.hpp`,

la implementació de l'arbre binari `arbreBin.hpp` i el programa principal `program.cpp`.

Exemple d'entrada 1

```
6
2 0
4 0
3 2
6 0
8 -1
5 2
```

Exemple de sortida 1

```
[5]
 \_[8]
 |  \_.
 |  \_[6]
 |  \_.
 |  \_.
 \_[3]
   \_[4]
   |  \_.
   |  \_.
   \_[2]
     \_.
     \_.
```

sí

Exemple d'entrada 2

```
6
2 0
4 0
3 2
9 0
8 1
5 2
```

Exemple de sortida 2

```
[5]
 \_[8]
 |  \_[9]
 |  |  \_.
 |  |  \_.
 |  \_.
 \_[3]
   \_[4]
   |  \_.
   |  \_.
   \_[2]
     \_.
     \_.
```

NO

Informació del problema

Autor : PRO1-Vilanova
Generació : 2021-05-31 19:22:46

© *Jutge.org*, 2006–2021.
<https://jutge.org>