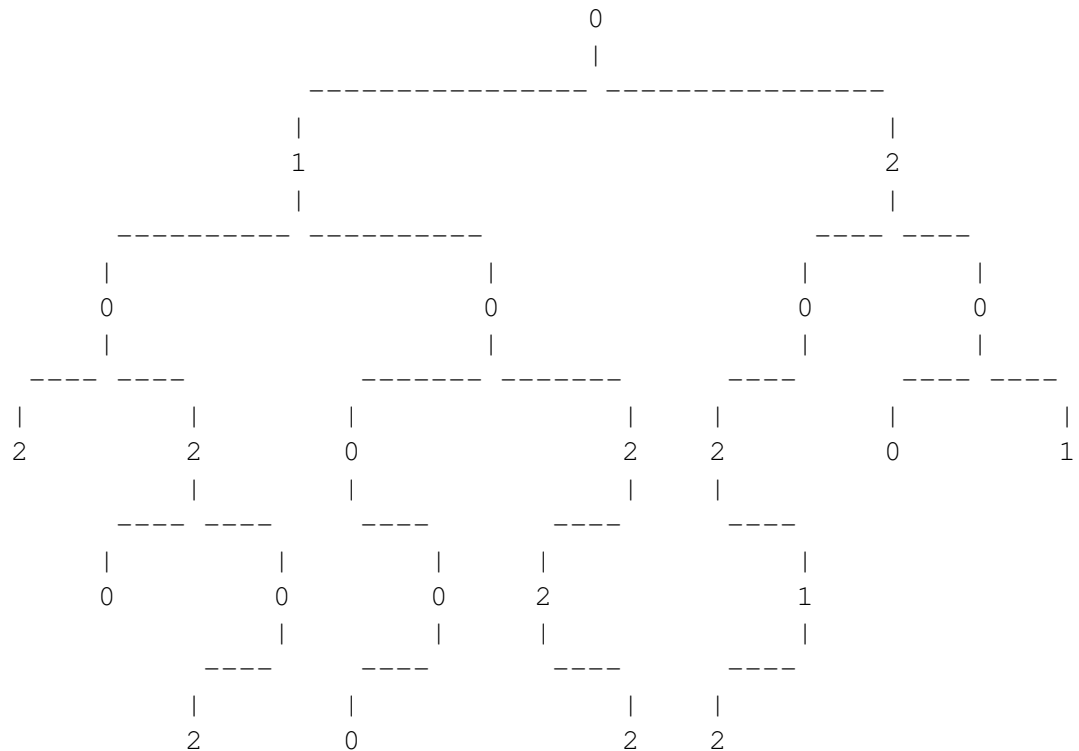


Mètode de la classe `Arbre` per a comptar el nombre de bessons `X69007_ca`

En aquest exercici afegirem un nou mètode `numTwins` a la classe `Arbre` per a calcular el nombre de parelles de nodes bessons, és a dir, que son germans (comparteixen el mateix node pare) i tenen el mateix valor.

Per exemple, suposeu que aquest és l'arbre representat per la variable `a` de tipus `Arbre`:



Llavors, la crida a `a.numTwins()` ha de retornar 4.

D'entre els fitxers que s'adjunten en aquest exercici, trobareu `Arbre.hh`, a on hi ha una implementació de la classe genèrica `Arbre`. Haureu de buscar dins `Arbre.hh` les següents línies i implementar els mètodes que s'hi indiquen:

```

// Pre:
// Post: Retorna el nombre de parelles de nodes de l'arbre representat pel pa
// que tenen el mateix node pare i el mateix valor.
// Descomenteu les següents dues línies i implementeu el mètode:
// int numTwins() const{
// }
private:

// Pre:
// Post: Retorna el nombre de parelles de nodes de l'arbre representat per n
// que tenen el mateix node pare i el mateix valor.
// Descomenteu les següents dues línies i implementeu el mètode:
// static int numTwinsAux(node_arbre *n ){

```

```
// }
```

Podeu suposar que el tipus genèric `T` de la classe té predefinida la operació de comparació `==`. De fet, es testearà la vostra implementació amb el tipus `T=int`. Ara bé, una solució que no sigui genèrica es considerarà incorrecta i serà invalidada a posteriori, encara que superi els jocs de proves.

D'entre els fitxers que s'adjunten a l'exercici també hi ha `main.cc` (programa principal), i el podeu compilar directament, doncs fa `include` de `Arbre.hh`. Només cal que pugueu `Arbre.hh` al jutge.

Entrada

L'entrada conté un nombre arbitrari d'arbres. Cada cas consisteix en una descripció d'un arbre binari d'enters. La descripció consisteix en un recorregut en preordre dels nodes de l'arbre, amb marques on hi anirien els arbres buits. Fixeu-vos en que el programa que us oferim ja s'encarrega de llegir aquestes entrades. Només cal que implementeu els mètodes abans esmentats.

Sortida

Per a cada cas, la sortida conté la corresponent avaluació de l'arbre. Fixeu-vos en que el programa que us oferim ja s'encarrega d'escriure aquesta avaluació. Només cal que implementeu els mètodes abans esmentats.

Exemple d'entrada

```
0 1 0 # # 2 # # 2 # 0 # #
2 2 # # 0 # #
0 # 0 # 0 # #
2 # 2 # #
1 1 # 0 # 1 # # 0 # #
1 1 1 # 2 # # 2 # # 1 # 2 0 # # 2 # #
2 0 # 2 # 1 # # #
1 0 1 # # 0 0 # 2 # # 2 0 # # 2 # # 0 0 #
0 0 0 # # 0 2 1 # # 2 # # 0 # # 2 2 # # 0
1 1 # # 0 2 # # 0 2 2 0 # # # 2 0 # # 0 #
1 2 # # #
1 2 # 2 # # 0 2 0 # # 2 1 # # 1 # # 1 0 #
1 # 0 2 2 # # 2 # # 0 2 # # 2 # #
1 2 0 # # # 1 1 # # 2 # #
2 0 0 2 # # 0 # # # 0 1 0 # # # 2 2 # # #
2 0 # 1 # 1 # # #
2 0 2 # # # 0 1 2 2 # # 0 # # 1 0 # # 1 #
0 2 2 # # 0 # # 0 2 # # 2 # #
1 2 # # 0 0 0 1 # # 2 # # 0 0 # # # 2 2 1
2 1 # # 1 # #
1 2 # # 1 # #
1 0 # # 0 # #
2 0 # # 0 0 # 1 2 # # 0 # 2 # # 0 0 2 # #
2 # #
1 2 2 2 2 2 # # 1 # # 1 # # 0 0 1 # # 0 #
1 2 0 # 0 # # 0 # 0 # # 2 1 # # 2 # #
2 0 2 # # 1 1 # # 1 # # 0 2 # 0 # # 1 2 #
2 0 # # 1 # #
1 2 2 0 # # 1 # # 0 0 1 # 1 # # 0 # 1 # #
```

```

0 2 # # 0 # #
1 1 0 # # # #
0 2 # 1 0 2 # 1 # # # # 1 0 # 0 2 1 # # # 1
1 # 1 # #
0 1 # 0 0 # # 2 # # 2 2 # 0 # # 2 1 # # #
1 1 2 # 2 1 1 # # 2 # # 0 # # 2 1 1 # # # 1
2 0 0 # # 2 # # 2 # #
2 0 2 # # 2 # # #
1 1 1 1 1 2 # # 2 # # 2 2 # # # 2 0 # # # 2
2 0 1 0 2 # # # 0 # 2 # # 2 1 # # 1 # # # 1
1 2 # # #
2 1 0 2 # # 2 1 # # 1 # # 2 # 0 # # 0 # 2
1 1 # # 2 0 # # 0 # #
1 0 # 0 # # 1 # #
0 2 0 1 # 1 # # # 0 0 # # # 0 0 2 # # 0 1
1 1 2 # # 0 # # 2 0 2 0 # # 1 # # 0 1 # #
2 1 2 # # 1 # # 2 1 1 # # 2 # # 0 # #
0 2 1 0 0 0 # # 2 # # 0 0 # # 1 # # 1 0 1
0 0 # # #
0 1 # # 1 2 1 2 # # # # 0 # 2 # 1 # #

```

Exemple de sortida

```

0 # 0 2 # # # #
0
0
0 # # 2 1 # 0 1 # 0 # # # 1 1 0 # 2 # # 2 0 # # 1 # #
0
1
0 # 2 2 # 1 # # 1 # 0 2 # # 1 # # 1 1 2 2 # # # 2 # #
1 0 # 2 # # 2 2 # # 1 # # 1 2 1 # # 2 # # 0 1 # # 2 #
1
4 # # 1 # #
0
2
2 # # 1 1 1 # # 1 # # 0 # 2 # #
0 # # #
1
0 # 0 # # 1 # # 2 2 0 # 1 # # # 2 # 0 0 # # # 2 1 0 0
2
1
2
1
0
1
4
0
0
2
2
0
4
1
1
4
0
2
0
1
0
0
1
0
0
1
2
1
1
3
3
1
0
2
0
1
0
0
1
2
1
1
1
0
0
0

```

0	0
0	2
0	1
1	0
0	2
1	0
4	0
0	3
1	0
3	1
2	

Observació

Avaluació sobre 10 punts: (Afegiu comentaris si el vostre codi no és prou clar)

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta, de cost lineal i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

Una solució que no sigui genèrica (per a qualsevol tipus T amb $==$) serà invalidada i rebrà nota 0, encara que superi els jocs de proves.

Informació del problema

Autor : PRO2

Generació : 2024-04-26 18:55:32

© *Jutge.org*, 2006–2024.

<https://jutge.org>