
Adaptar BinaryTree per a mantenir informació sobre la midaX30464_ca

En aquest exercici haureu de modificar `BinaryTree` amb la finalitat de que els arbres també mantinguin informació correcta sobre la seva pròpia mida. En particular, haureu d'afegir un nou mètode `getSize` que retorni el nombre de nodes de l'arbre. Aquesta operació hauria de tenir cost constant.

D'entre els fitxers que s'adjunten en aquest exercici, trobareu `BinaryTree.old.hpp`, a on hi ha una implementació de la classe genèrica `BinaryTree`. En primer lloc, haureu de fer:

```
cp BinaryTree.old.hpp BinaryTree.hpp
```

A continuació, heu de fer tot un seguit de canvis:

- Heu d'afegir un nou atribut `int size`.
- Heu d'afegir un nou mètode privat per a actualitzar la mida de l'arbre i la mida dels seus antecessors (els arbres que tenen a l'arbre actual com a subarbre). Això es pot fer de forma recursiva o iterativa. Una possible manera iterativa és:

```
void updateSize()
{
    BinaryTree<T> *pt = this;
    while (pt != NULL) {
        if (pt->isEmpty()) pt->size = ...;
        else pt->size = ...;
        pt = pt->parent;
    }
}
```

Una possible manera recursiva és:

```
void updateSize()
{
    if (isEmpty()) size = ...;
    else size = ...;
    if (parent != NULL) parent->updateSize();
}
```

- A les constructores i a l'operació d'assignació heu d'afegir crides a `updateSize`.
- Heu d'implementar el mètode `getSize`.

D'entre els fitxers que s'adjunten a l'exercici també hi ha `program.cpp` (programa principal) i `Makefile` per a compilar. Per a pujar la vostra solució, heu de crear el fitxer `solution.tar` així:

```
tar cf solution.tar BinaryTree.hpp
```

Entrada

El programa principal té una variable d'arbre d'enters `t`, inicialment buida, i llegeix instruccions que, o bé mostren com és `t`, o bé modifiquen algun subarbre de `t` o mostren la mida d'algun subarbre de `t`. Les instruccions que mostren `t` són simplement de la forma `<< t`. Les altres instruccions comencen per `t`, seguit d'una seqüència de `.left` o `.right`. Finalment, o bé la instrucció acaba amb `.size`, cas en el qual s'escriurà la mida del corresponent subarbre, o ve seguida de `= t'`, on `t'` és un string que representa un arbre, cas en el qual `t'` (com a arbre) serà assignat al corresponent subarbre de `t`. Per exemple:

```
t = 3(4, 5(1, 2))
<< t
t.size
t.left.size
t.right.size
t.right.left = 8(9, 10)
<< t
t.right.size
```

La sortida de la seqüència anterior és:

```
3(4, 5(1, 2))
5
1
3
3(4, 5(8(9, 10), 2))
5
```

Com podeu observar, el `size` d'un arbre que està per sobre del que hem assignat també ha estat actualitzat.

Se suposa que la seqüència d'entrada serà correcta (sense accessos fora de l'arbre, tot i que sí que es pot accedir a subarbres buits de l'arbre).

El programa principal que us oferim ja s'encarrega de llegir aquestes entrades i fer les crides als corresponents mètodes de la classe `BinaryTree`. Només cal que feu les modificacions abans esmentades dins el fitxer `BinaryTree.hpp`.

Sortida

Per a cada instrucció `<< t`, s'escriurà el contingut actual de l'arbre. Per a cada instrucció acabada en `size`, s'escriurà la mida del subarbre indicat. El programa que us oferim ja fa això. Només cal que feu les modificacions abans esmentades dins el fitxer `BinaryTree.hpp`.

Exemple d'entrada 1

```
t = 7(2, 5)
t.size
<< t
t = 5(, 1)
t.size
<< t
t.left = 4(2(, 3), 2)
t.left.size
<< t
```

```
t.left.left = 7(3,)
t.left.size
<< t
t.right = 5(6(1,), 2)
t.size
<< t
t.left.right = 5(, 8(, 3))
t.size
<< t
t.left.right.left = 1(3, 4(3, 2))
```

```
t.size
<< t
t.right.right = 2(5(2,2),)
t.right.right.size
<< t
t.left.left = 6
t.left.left.size
<< t
t.right.right.left = 1
t.right.size
<< t
```

Exemple de sortida 1

```
3
7(2,5)
2
5(,1)
4
5(4(2(,3),2),1)
4
5(4(7(3,),2),1)
9
5(4(7(3,),2),5(6(1,),2))
11
5(4(7(3,),5(8(,3))),5(6(1,),2))
16
5(4(7(3,),5(1(3,4(3,2)),8(,3))),5(6(1,),2))
4
5(4(7(3,),5(1(3,4(3,2)),8(,3))),5(6(1,),2(5(2,2),)))
1
5(4(6,5(1(3,4(3,2)),8(,3))),5(6(1,),2(5(2,2),)))
5
5(4(6,5(1(3,4(3,2)),8(,3))),5(6(1,),2(1,)))
```

Informació del problema

Autoria: PRO1

Generació: 2026-01-25T21:07:40.640Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>