
Arbre general. Nivell del primer element x en post-ordre

Donada la classe *Arbre* que permet gestionar arbres generals usant memòria dinàmica, cal implementar el mètode

```
int nivell_postordre (const T &x) const;
```

que retorna el nivell a on està el primer element x fent un recorregut en post-ordre. Retorna -1 si x no hi és.

Cal enviar a jutge.org la següent especificació de la classe *Arbre* i la implementació del mètode dins del mateix fitxer. Indica dins d'un comentari a la capçalera del mètode el seu cost en funció del nombre d'elements n de l'arbre.

```
#include <iostream>
```

```
using namespace std;
```

```
template <typename T>
```

```
class Arbre {
```

```
public:
```

```
// Pre: True
```

```
// Post: Crea un Arbre format per un únic node que conté a x.
```

```
Arbre(const T &x);
```

```
// Tres grans.
```

```
Arbre(const Arbre<T> &a);
```

```
Arbre& operator=(const Arbre<T> &a);
```

```
~Arbre() throw();
```

```
// Pre: True
```

```
// Post: L'Arbre a ha quedat com a darrer fill de l'arrel de l'arbre sobre el que s'aplica el mètode i l'arbre a queda invalidat;
```

```
// després de fer b.afegir_fill(a), a no és un arbre vàlid.
```

```
void afegir_darrer_fill (Arbre<T> &a);
```

```
static const int ArbreInvalid = 400;
```

```
// Pre: True
```

```
// Post: Retorna el nivell a on està el primer element  $x$  fent un recorregut en post-ordre.
```

```
// Retorna -1 si  $x$  no hi és.
```

```
int nivell_postordre (const T &x) const;
```

```
private:
```

```
Arbre(): _arrel ( nullptr ) {};
```

```
struct node {
```

```
    T info;
```

```
    node* primf;
```

```
    node* seggerm;
```

```

};
node* _arrel ;
static node* copia_arbre (node* p);
static void destrueix_arbre (node* p) throw();

// Aquí va l'especificació dels mètodes privats addicionals
};

// Aquí va la implementació del mètode nivell_postordre i dels privats addicionals

```

Per testejar la solució, jutge.org ja té implementats la resta de mètodes de la classe *Arbre* i un programa principal que llegeix un arbre general d'enters i després crida el mètode *nivell_postordre*.

Entrada

L'entrada consisteix en la descripció d'un arbre general d'enters (el seu recorregut en preordre, en el qual al valor de cada node li segueix el seu nombre de fills). A continuació segueix una seqüència d'enters que representen diferents valors per testejar *nivell_postordre*.

Sortida

Una línia per cada element x de la seqüència d'enters d'entrada, amb el nivell a on està el primer element x fent un recorregut en post-ordre (-1 si x no hi és).

Observació

Només cal enviar la classe requerida i la implementació del mètode *nivell_postordre* amb el seu cost en funció del nombre d'elements n de l'arbre. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat.

Exemple d'entrada 1

```

7 0
9
7

```

Exemple de sortida 1

```

-1
0

```

Exemple d'entrada 2

```

7 2
  8 0
  -8 0
9
7
8
-8

```

Exemple de sortida 2

```

-1
0
1
1

```

Exemple d'entrada 3

```

-7 3
  8 0
  4 2
    3 1
      0 1

```

```

        6 0
      -5 0
    2 4
      9 0
      1 0
      2 0
      5 0

```

0
1
2
3
4
5
6
7
8
9
-7

Exemple d'entrada 4

-7 3
8 0
4 2
2 1
1 1
9 0
-7 1
2 0
2 4
9 0
1 0
8 0
4 0
0
1
2
3
4
5
6
7
8
9
-7

Exemple de sortida 3

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

Exemple de sortida 4

-1
3
2
-1
1
-1
-1
-1
1
4
2

Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T15:57:43.098Z

© Jutge.org, 2006–2026.

<https://jutge.org>