
Arbre general. Crea fills amb els màxims dels elements de cada camí X22358_ca

Donada la classe *Arbre* que permet gestionar arbres generals usant memòria dinàmica i un tipus *T* que disposa d'operadors de comparació, cal implementar el mètode

```
void crea_fills_maxims_camins ();
```

que per cada fulla afegeix un nou fill que conté el màxim de tots els elements del camí que va de l'arrel a la fulla.

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 <cstdlib>
#include <string>
using namespace std;
typedef unsigned int nat;
```

```
template <typename T>
class Arbre {
```

public:

```
// Construeix 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();
```

```
// Col·loca l'Arbre donat 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);
```

```
// Imprimeix la informació dels nodes en preordre, cada element en una nova línia i
precedit per espais segons el nivell on està situat.
```

```
void preordre () const;
```

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

```
// Per cada fulla afegeix un nou fill que conté el màxim de tots els elements del camí
que va de l'arrel a la fulla.
```

```
void crea_fills_maxims_camins ();
```

private:

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

```
struct node {
    T info;
```

```

    node* primf;
    node* seggerm;
};
node* _arrel ;
static node* copia_arbre (node* p);
static void destrueix_arbre (node* p) throw();
static void preordre (node *p, string pre);

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

// Aquí va la implementació del mètode crea_fills_maxims_camins i dels privats addi-
cionals

```

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 *crea_fills_maxims_camins*.

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).

Sortida

El recorregut en preordre de l'arbre general resultant. Cada element en una nova línia i precedit per espais segons el nivell on està situat.

Observació

Només cal enviar la classe requerida i la implementació del mètode *crea_fills_maxims_camins* 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
```

Exemple d'entrada 2

```
7 2
 8 0
-8 0
```

Exemple d'entrada 3

```
-7 3
 8 0
 4 2
 3 1
```

Exemple de sortida 1

```
7
 7
```

Exemple de sortida 2

```
7
 8
 8
-8
 7
```

```

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

8 0
5 0

Exemple de sortida 3

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

Exemple d'entrada 4

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

Exemple de sortida 4

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

Informació del problema

Autor : Jordi Esteve

Generació : 2021-10-27 18:14:22

© *Jutge.org*, 2006–2021.

<https://jutge.org>