Arbre general. Modifica arbre amb altures de cada subarbre.X23363_ca

Donada la classe *Arbre* que permet gestionar arbres generals d'enters usant memòria dinàmica implementats amb la tècnica de primer fill-següent germà, cal implementar el mètode

```
void arbre_altures ();
```

que modifica el contingut dels nodes per tal de guardar a cada node l'altura del seu subarbre. Cal enviar a jutge.org la següent especificació de la classe *Arbre* i la implementació del mètode dins del mateix fitxer. Al principi de cada mètode implementat, dins d'un comentari, cal indicar el cost temporal amb el raonament corresponent, incloent l'equació de la recurrència si fos necessari.

```
#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 primer 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_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;

  // Modifica el contingut dels nodes per tal de guardar a cada node l'altura
  // del seu subarbre
  void arbre_altures ();

private:
  Arbre(): _arrel(NULL) {};
  struct node {
    T info;          // Informació del node
```

```
    node* primf;     // Punter al primer fill
    node* seggerm;  // Punter al següent germà
  };
  node* _arrel ;  // Punter a l'arrel de l'arbre
  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 arbre_altures i 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 els mètodes *arbre_altures* i *preordre*.

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 *arbre_altures*. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat. Al principi de cada mètode implementat i dins d'un comentari cal indicar el cost temporal amb el raonament corresponent, incloent l'equació de la recurrència si fos necessari.

Exemple d'entrada 1

```
7 0
```

Exemple de sortida 1

```
1
```

Exemple d'entrada 2

```
7 1
  8 0
```

Exemple de sortida 2

```
2
  1
```

Exemple d'entrada 3

```
7 2
  8 0
  9 1
    4 0
```

Exemple de sortida 3

```
3
  1
  2
    1
```

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

```
6
  4
    3
      2
        1
        1
        1
  5
    4
      3
        2
          1
        1
        2
          1
          1
          1
        1
        2
          1
          1
    1
```

Exemple d'entrada 5

```
7 4
  -5 0
  -9 1
    8 0
  -3 0
  -4 0
```

Exemple de sortida 5

```
3
  1
  2
    1
  1
  1
```

Informació del problema

Autoria: Jordi Esteve

Generació: 2026-01-25T14:28:02.172Z