

---

## Desequilibris a l'inserir una clau en un BST

X86307\_ca

---

Donada la classe *dicc* que permet gestionar diccionaris on només hi guardem claus úniques usant arbres binaris de cerca (BST), cal implementar el mètode

```
vector<Clau> insereix(const Clau &k);
```

que insereix la clau *k* en el diccionari si no hi era i sempre retorna un vector amb les claus dels nodes visitats on hi ha un desequilibri després de la inserció, començant per la més propera a les fulles i acabant per la més propera a l'arrel. Un node està desequilibrat quan el factor d'equilibri és superior a 1.

ATENCIÓ: No cal equilibrar el BST tal com succeeix amb els AVL, sinó que dels nodes visitats només cal indicar els que no compleixen que el factor d'equilibri sigui igual o inferior a 1. Un BST pot tenir varis nodes desequilibrats.

Les claus són del tipus *Clau* que admet una relació d'ordre total, és a dir, tenim una operació de comparació *<* entre claus.

Cal enviar a jutge.org la següent especificació de la classe *dicc* 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 del diccionari en el cas mig i en el cas pitjor.

```
#include <iostream>
#include <vector>
using namespace std;
typedef unsigned int nat;
```

```
template <typename Clau>
```

```
class dicc {
```

```
public:
```

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

```
    // Pre: Cert
```

```
    // Post: El resultat és un dicc sense cap element
```

```
    ~dicc ();
```

```
    // Pre: Cert
```

```
    // Post: El dicc ha estat destruït
```

```
    vector<Clau> insereix(const Clau &k);
```

```
    // Pre: Cert
```

```
    // Post: La clau k s'ha inserit en el diccionari si no hi era. Retorna un vector amb
```

```
    // les claus dels nodes visitats on hi ha un desequilibri després de la inserció,
```

```
    // començant per la més propera a les fulles i acabant per la més propera a l'arrel.
```

```
private:
```

```
    struct node {
```

```
        Clau _k;        // Clau
```

```
        node* _esq;    // fill esquerre
```

```
        node* _dret;   // fill dret
```

```
        nat _h;        // Altura del subarbre
```

```

};
node * _arrel ;

static void esborra_nodes (node* m);

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

// Aquí va la implementació dels mètodes públics i privats

```

Degut a que jutge.org només permet l'enviament d'un fitxer amb la solució del problema, en el mateix fitxer hi ha d'haver l'especificació de la classe i la implementació del mètode *insereix* (el que normalment estarien separats en els fitxers *.hpp* i *.cpp*). Per testejar la classe disposes d'un programa principal que llegeix i insereix enters a un diccionari amb claus enters inicialment buit.

## Entrada

L'entrada conté una seqüència d'enters, són els elements que s'insereixen a un diccionari inicialment buit.

## Sortida

Per a cada enter d'entrada, escriu una línia amb el text "insereix", seguit de l'enter d'entrada, el separador ":" i la llista de claus enters a on s'han detectat desequilibris separades per un espai.

## Observació

Només cal enviar la classe requerida i la implementació del mètode *insereix*. Pots ampliar la classe amb mètodes privats. Segueix estrictament la definició de la classe de l'enunciat. Has de visitar els nodes del BST estrictament necessaris per fer la inserció, per això et pots ajudar de l'atribut *h* de cada node que guarda l'altura del seu subarbre. Aquest atribut l'has d'actualitzar quan insereixis una clau. Indica dins d'un comentari a la capçalera del mètode el seu cost en funció del nombre d'elements del diccionari en el cas mig i en el cas pitjor.

### Exemple d'entrada 1

```

6
8
9
7
4

```

### Exemple d'entrada 2

```

5
-3
8
2
-1
-3

```

### Exemple de sortida 1

```

insereix 6:
insereix 8:
insereix 9: 6
insereix 7: 6
insereix 4:

```

```

7
-7
6
-2
9
0
-4

```

## Exemple de sortida 2

```
insereix 5:  
insereix -3:  
insereix 8:  
insereix 2:  
insereix -1: -3 5
```

```
insereix -3: -3 5  
insereix 7:  
insereix -7:  
insereix 6: 8  
insereix -2: 2 -3  
insereix 9:  
insereix 0: 2 -3  
insereix -4:
```

## Informació del problema

Autor : Jordi Esteve

Generació : 2021-01-25 11:43:10

© *Jutge.org*, 2006–2021.

<https://jutge.org>