



Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PUBLICACIÓ DOCENT

MANUAL DE LABORATORI D'ESIN Sessió 6

AUTOR: Bernardino Casas, Jordi Esteve

ASSIGNATURA: Estructura de la Informació (ESIN)

CURS: Q3

TITULACIONS: Grau en Informàtica

DEPARTAMENT: Ciències de la Computació

ANY: 2020

Vilanova i la Geltrú, 12 d'octubre de 2020

6

Exercici

L'objectiu d'aquest exercici és resoldre problemes d'arbres generals usant un arbre general implementat per nosaltres mateixos usant memòria dinàmica.

Caldrà resoldre els següents problemes de la plataforma jutge.org; els trobaràs en l'apartat Arbres del curs ESIN (Vilanova):

- [P66413](#). Recorreguts recursius d'un arbre general.
- [P66413](#). Suma d'arbres.

No has de guardar l'arbre en un vector tal com s'explica en l'enunciat del primer problema, sinó que has d'usar una classe que implementi els arbres generals en memòria dinàmica. L'especificació i implementació d'arbre general en memòria dinàmica la trobaràs en els apunts de teoria; per evitar problemes copiant des de fitxers PDF la pots copiar de la carpeta `/home/public/esin/sessio6`).

A més a més de l'especificació i implementació de la classe `Arbre` (fitxers `arbre.hpp` i `arbre.t`), cal implementar altres funcions i el programa principal que les cridarà (`main.cpp`).

En aquest exercici, el mateix fitxer hi ha d'haver l'especificació i la implementació de la classe i la resta de funcions i programa principal. I també cal eliminar les directives `#include "arbre.t"` i `#include "arbre.hpp"` per no tenir problemes de precompilació. Ho pots fer tot a la vegada amb la comanda:

```
cat arbre.hpp arbre.t main.cpp | sed '/include "arbre./d' > solucio.cpp
```

i enviar a jutge.org el fitxer `solucio.cpp`.

6.1 Consells

Pots solucionar aquests problemes fent funcions externes a la classe `Arbre` o fent nous mètodes dins de la classe `Arbre`:

- Si fas funcions externes a la classe `Arbre`, només pots accedir als mètodes públics de la classe, per tant hauràs d'usar iteradors per recórrer l'arbre general.
- Si fas nous mètodes dins de la classe `Arbre`, aquests també poden usar iteradors per recórrer l'arbre general. Però com que podem accedir directament als atributs privats, una solució més eficient i elegant és usar punters a node i visitar els nodes de l'arbre general seguint els enllaços primer fill i següent germà.

Et recomanem la primera opció per fer el primer problema; fixa't que és un problema molt semblant al fet a la sessió 5, canviant arbres binaris per arbres generals.

I et recomanem la segona opció per fer el segon problema, així practiques amb la implementació amb punters dels arbres generals. Per exemple podríes fer el mètode públic

```
Arbre operator+(const Arbre<T> &a);  
// Retorna un arbre que és la suma del p.i. i l'arbre a
```

que crida al mètode privat recursiu

```
static node* suma_arbres(node *a1, node *a2);  
// Retorna el punter a l'arrel de la suma dels arbres apuntats per a1 i a2
```

I fer el mètode públic

```
void preordre() const;  
// Escriu una línia amb el recorregut en preordre de l'arbre general  
// Cada element ha de sortir precedit d'un espai.
```

que crida al mètode privat recursiu

```
static void preordre(node *n);  
// Escriu una línia amb el recorregut en preordre de l'arbre general  
// que apunta n. Cada element ha de sortir precedit d'un espai.
```

6.2 Exercicis opcionals

Pots practicar fent la resta dels exercicis de l'apartat Arbres del curs ESIN (Vilanova).

Tingues en compte que per resoldre alguns problemes no és necessari reconstruir l'arbre, ja que es pot anar calculant el resultat a mida que es llegeix el recorregut en preordre de l'entrada. Per tant, en aquests casos, no caldrà incloure la classe `Arbre` dins del nostre codi.