



Escola Politècnica Superior  
d'Enginyeria de Vilanova i la Geltrú

UNIVERSITAT POLITÈCNICA DE CATALUNYA

## PUBLICACIÓ DOCENT

# MANUAL DE LABORATORI D'ESIN Sessió 7

**AUTOR:** Bernardino Casas, Jordi Esteve

**ASSIGNATURA:** Estructura de la Informació (ESIN)

**CURS:** Q3

**TITULACIONS:** Grau en Informàtica

**DEPARTAMENT:** Ciències de la Computació

**ANY:** 2021

Vilanova i la Geltrú, 12 d'octubre de 2021



# 7

## Exercici

### 7.1 Operacions bàsiques de BSTs

---

L'objectiu d'aquest exercici és resoldre problemes d'arbres binaris de cerca (ABC o BST) usant un arbre binari implementat per nosaltres mateixos usant memòria dinàmica.

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

- [P94818](#) Inserint en un arbre de cerca.
- [P92765](#) Buscant en un arbre de cerca

Pots resoldre aquests exercicis usant un arbre binari en memòria dinàmica i afegint mètodes per inserir i cercar elements considerant que l'arbre binari és de cerca (és un ABC o BST). Recorda que els arbres binaris els vam practicar a la sessió 5 i la seva especificació i implementació la pots copiar de la carpeta `/home/public/esin/sessio5`.

#### 7.1.1 Consells

Per resoldre aquests dos problemes, a la classe `Abin` es podria afegir el mètode

```
void insereix_bst(const T &k);  
// Pre: cert  
// Post: L'element k s'ha inserit en el p.i. tenint en compte que és un BST
```

que crida al mètode privat recursiu

```
static node* insereix_bst(node *n, const T &k);
```

i el mètode

```
bool consulta_bst(const T &k) const;  
// Pre: cert  
// Post: Retorna true si l'element k està en el p.i. tenint en compte que és un BST
```

que crida al mètode privat recursiu

```
static node* consulta_bst(node *n, const T &k);
```

En la sortida del primer problema, fixa't que cada element de l'arbre binari s'imprimeix en una nova línia, ocupa 10 caràcters i li precedeixen *nivell* \* 10 espais en blanc.

## 7.2 Diccionaris. Cerca eficient per rang i per posició

---

Volem resoldre el problema de cerca eficient en diccionaris, on a més de cercar una clau concreta volem cercar per rang (entre dues claus  $k_1$  i  $k_2$ ), per posició (clau de la posició  $i$ -èsima), la clau mínima i la clau màxima.

Caldrà resoldre aquest problema de la plataforma jutge.org; el trobaràs en l'apartat Diccionaris del curs ESIN (Vilanova):

- [X68607](#). Classe Diccionari (I).