

---

## Cerca en un arbre binari de parelles

S70309\_ca

---

Implementa la funció:

```
/**
 * @brief Cerca l'enter d'un `pair<int, string>` en un arbre binari i
 *        retorna el `string` associat a aquest enter.
 *
 * @param t Arbre binari de parelles d'enters i `string`s.
 * @param x Valor a cercar.
 * @param trobat Paràmetre de sortida amb l'`string` associat a `x`.
 *
 * @returns `true` si `x` és el `first` d'algun parell a `t`,
 *          `false` en cas contrari.
 *
 * @post Si el resultat és `true`, `trobat` conté l'`string` associat a `x`.
 */
bool find_first(BinTree<pair<int, string>> t, int x, string& trobat);
```

### Observació

	bintree.hh	la classe BinTree
Els fitxers públics (icona del gatet) contenen:	bintree-io.hh	l'entrada/sortida de BinTree (inclou op
	main.cc	el programa principal

També hi ha un Makefile i el directori .vscode que té la configuració per compilar i debuggar amb VSCode.

Cal implementar `find_first` en un **fitxer .cc nou**, compilar (està preparat per poder compilar i debuggar amb VSCode), i finalment **enviar només el fitxer amb la funció**.

La solució ha de ser **eficient**: cal que abandoni la cerca tan aviat com trobi el valor buscat.

### Entrada

L'entrada conté un arbre binari de parelles `pair<int, string>` en format visual, seguit d'una línia buida, i tot seguit una seqüència d'enters a cercar.

### Sortida

Per a cada enter cercat, si es troba a l'arbre, s'escriu l'enter i l'string associat. Si no es troba, s'escriu un missatge indicant-ho.

### Exemple d'entrada

```
1 alpha
|-- 2 bravo
|   |-- 10 juliet
|   '-- 8 hotel
'-- 3 charlie
```

```
1 2 3 4 5 6 7 8 9 10
```

## Exemple de sortida

1 alpha  
2 bravo  
3 charlie  
No s'ha trobat 4

No s'ha trobat 5  
No s'ha trobat 6  
No s'ha trobat 7  
8 hotel  
No s'ha trobat 9  
10 juliet

## Informació del problema

Autoria: Pau Fernández

Generació: 2026-04-02T20:59:21.541Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>