
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

Els fitxers públics (icona del gatet) són: la classe `BinTree` (fitxer `bintree.hh`), l'entrada/sortida de `BinTree` (`bintree-io.hh`) i el programa principal. També hi ha un `Makefile` i el directori `.vscode` que té la configuració per compilar i debuggar amb VSCode.

Has d'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**.

Els exemples d'entrada/sortida mostren que el `main` que utilitza el Jutge llegeix un arbre de parelles i tot seguit una seqüència d'enters a cercar i, per a cadascún, crida la funció `find_first`, mostrant el resultat de la cerca. (Per tant, no cal implementar-lo.)

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

Autor : Pau Fernández

Generació : 2025-03-04 18:35:51

© Jutge.org, 2006–2025.

<https://jutge.org>