

---

**Cerca en un arbre de parells d'enters****X20646\_ca**

---

Heu d'implementar una cerca en un arbre de parells d'enters. Com a entrada hi haurà els parells d'un arbre binari en preordre: primer l'arrel, després el subarbre esquerre i per últim el dret. Els arbres buits es representen amb el parell (0, 0). Tot seguit hi haurà una llista d'enters. Per a cadascun dels enters cal cercar-lo en l'arbre en la primera posició del parell. Si hi és, cal dir (en aquest ordre) l'element que cerquem, quin és el seu company en el parell, i en quina profunditat es troba (assumiu que la profunditat de l'arrel és 0). Si no hi és, cal treure un -1.

**Entrada**

L'entrada és un arbre binari de parells en preordre, sense repeticions respecte al primer element de cada parell, i una seqüència d'enters per cercar.

**Sortida**

Per a cada enter, la sortida és -1 si l'enter no és a la primera posició d'un parell a l'arbre, altrament, cal treure l'element, el seu company i la profunditat en què es troba.

**Observació**

Cal fer servir la classe `BinTree` que us donem

Heu d'enviar tres fitxers en un sol .tar.

- `BinTreeIOParInt.hh` amb les funcions:

```
void read_bintree_parint(BinTree<ParInt>& a);
// Pre: a és buit; el canal estandar d'entrada conté un nombre
// parell d'enters, que representa un arbre binari en preordre,
// on el parell 0 0 representa un arbre buit
// Post: a conté l'arbre que hi havia al canal estandar d'entrada

void write_bintree_parint(const BinTree<ParInt>& a); (opcional)
// Pre: a = A
// Post: s'han escrit al canal estandar de sortida els elements
// d'a recorreguts en inordre, a = A
```
- `BinTreeIOParInt.cc` amb la seva codificació
- `program.cc` amb el programa

Observeu que per als parells d'enters us donem la classe `ParInt` que detecta si el parell llegit és 0 0 i que per compilar us donem el `Makefile`.

**Informació del problema**

Autor : Jaume Baixeries Juan Luis Esteban Borja Valles

Generació : 2020-10-06 18:08:28

© Jutge.org, 2006–2020.

<https://jutge.org>