

Camí preferent d'un arbre (BinTree)

X78332_ca

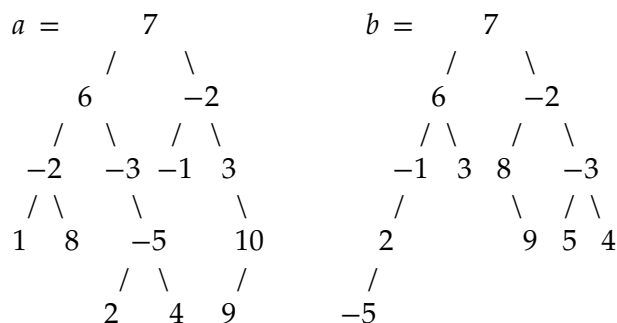
Recordeu que una *fulla* d'un arbre és un node sense cap successor. Un *camí* dins d'un arbre és una successió de nodes que van de l'arrel a una fulla.

Donat un BinTree **a** d'elements de qualsevol tipus, definim el *camí preferent* del BinTree **a** de la següent manera: si **a** és buit llavors el camí preferent d'**a** també és buit; en cas contrari, el camí preferent d'**a** el forma l'element arrel d'**a** seguit del camí preferent del fill d'**a** que tingui més elements. Si **a** té dos fills no buits amb el mateix nombre d'elements es tria el camí preferent del fill esquerre.

Volem una operació que ens permeti saber quin és el camí preferent d'un BinTree d'enters, representant aquest camí amb una pila d'enters, ordenada de forma que el primer element del camí (si existeix) sigui al cim de la pila. Feu només un recorregut de l'arbre. Eviteu còpies i assignacions de stacks i minimitzeu l'ús d'espai addicional. Feu servir la següent especificació:

```
void cami_preferent (const BinTree<int>& a, stack<int>& c)
/* Pre: c es buida */
/* Post: c conte el camí preferent d'a; si no es buit, el primer element
del camí es al cim de c */
```

Exemple: considereu els dos BinTree següents



- el camí preferent d'**a** és 7 6 -3 -5 2.
- el camí preferent de **b** és 7 -2 -3 5.

Entrada

L'entrada és un BinTree d'enters.

Sortida

La sortida és una pila amb el camí preferent. L'arrel de l'arbre és al cim de la pila.

Observació

Només s'ha d'enviar un fitxer que contengui la funció amb la capçalera de l'enunciat i qualsevol altra funció auxiliar que cregueu convenient, sense la funció main. Afegiu-hi també els includes de les classes BinTree i stack mitjançant

```
#include "BinTree.hh"
```

```
#include <stack>
```

Informació del problema

Autoria: Alberto Moreno (adapter), Borja Valles (responsable)

Generació: 2026-01-25T21:27:43.194Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>