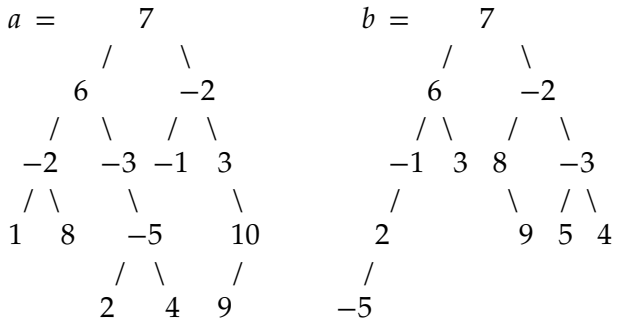
Camino preferente de un árbol (BinTree) X78332_es

Recordad que una *hoja* de un árbol es un nodo sin ningún sucesor. Un *camino* dentro de un árbol es una sucesión de nodos que van de la raíz a una hoja.
Dado un árbol binario **a** de elementos de cualquier tipo, definimos el *camino preferente* del árbol **a** de la siguiente manera: si **a** esta vacío entonces el camino preferente de **a** también esta vacío; en caso contrario, el camino preferente de **a** lo forman el elemento raíz de **a** seguido del camino preferente del hijo de **a** que tenga más elementos. Si **a** tiene dos hijos no vacíos con el mismo numero de elementos se elige el camino preferente del hijo izquierdo.
Queremos una operación que nos permita saber cual es el camino preferente de un árbol de enteros, representando este camino con una pila de enteros, ordenada de forma que el primer elemento del camino (si existe) esté en la cima de la pila. Recorred el árbol solamente una vez. Evitad las copias y asignaciones de stacks y minimizad el uso de espacio adicional. Utilizad la siguiente especificación:

```
void cami_preferent (const BinTree<int>& a, stack<int>& c)
/* Pre: c esta vacia */
/* Post: c contiene el camino preferente de a; si no esta vacia, el primer
         elemento del camino esta en la cima de c */
```

Ejemplo: considerad los dos árboles siguientes



- el camino preferente de **a** es 7 6 -3 -5 2.
- el camino preferente de **b** es 7 -2 -3 5.

Entrada

La entrada es un árbol de enteros.

Salida

La salida es una pila con el camino preferente. La raiz del árbol está en la cima de la pila.

Observación

Tan solo se debe enviar un fichero que contenga la función con la cabecera del enunciado y cualquier otra función auxiliar que creais conveniente, sin la función main. Añadid también los includes de las clases BinTree i stack mediante

#include "BinTree.hh"
#include <stack>

Información del problema

Autoría: Alberto Moreno (adapter), Borja Valles (responsible)

Generación: 2026-01-25T21:27:39.288Z