

Camino preferente de un árbol

X38658_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** está vacío entonces el camino preferente de **a** también está 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 número de elementos se elige el camino preferente del hijo izquierdo.

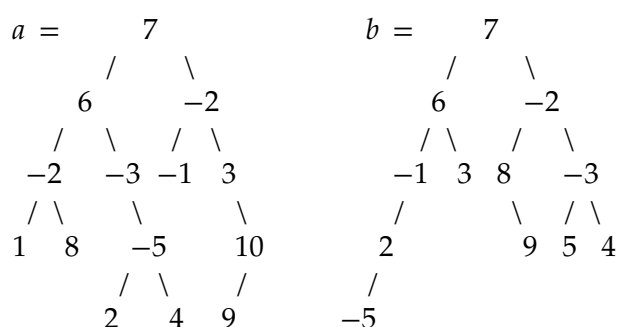
Queremos una operación que nos permita saber cuál es el camino preferente de un árbol de enteros, representando este camino con una pila de enteros, ordenada de forma que el primer elemento del camino (si existe) esté en la cima de la pila. Utilizad la siguiente especificación:

```
void cami_preferent (Arbre<int> &a, stack<int> &c)
```

```
/* Pre: a=A, c está vacía */
```

```
/* Post: c contiene el camino preferente de A; si no está vacía, el primer  
         elemento del camino está 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 raíz 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 creáis conveniente, sin la función main. Añadid también los includes de las clases Arbre i stack mediante

```
#include "Arbre.hh"
```

```
#include <stack>
```

Información del problema

Autoría: Alberto Moreno (adapter), Ramon Ferrer i Cancho (responsable)

Generación: 2026-01-25T15:48:26.105Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>