

Inserción en un BST

Y74034_es

Dado un árbol binario de búsqueda (BST) de enteros y un valor, implementa una función que devuelva un nuevo BST con el valor insertado en la posición correcta. Si el valor ya existe en el árbol, la función debe devolver el mismo árbol sin cambios.

```
/**
 * @brief Inserta un valor en un BST.
 *
 * @param t Un árbol binario de búsqueda.
 * @param x Valor a insertar.
 * @returns Un nuevo BST con `x` insertado en la posición correcta.
 *          Si `x` ya está en `t`, devuelve `t` sin cambios.
 */
BinTree<int> bst_inserta(BinTree<int> t, int x);
```

Recordad que `BinTree<int>` es inmutable: no se puede modificar un árbol existente, sino que hay que construir uno nuevo.

Entrada

La entrada consiste en una secuencia de casos. Cada caso contiene un árbol binario de búsqueda en formato visual y, a continuación, una secuencia de enteros a insertar.

Salida

Para cada caso, la salida muestra el árbol resultante de insertar todos los valores en el árbol inicial, en formato visual.

Observación

Los ficheros públicos (icono del gatito) son: la clase `BinTree` (fichero `bintree.hh`), la entrada/salida de `BinTree` (`bintree-io.hh`) y el programa principal. También hay un `Makefile` y el directorio `.vscode` con la configuración para compilar y depurar con VSCode.

Debes implementar `bst_inserta` en un **fichero .cc nuevo**, compilar (está preparado para poder compilar y depurar con VSCode), y finalmente **enviar solo el fichero con la función**.

Ejemplo de entrada

```
visual
10
|-- 5
|   |-- 2
|   '--- 7
'--- 15

3 12 20

#
```

```
8 4 12 2 6 10 14
```

Ejemplo de salida

```
10
|-- 5
|  |-- 2
|  |  |-- #
|  |  |-- 3
|  |-- 7
|-- 15
    |-- 12
```

```
      |-- 20
8
|-- 4
|  |-- 2
|  |-- 6
|-- 12
    |-- 10
    |-- 14
```

Información del problema

Autoría: Pau Fernández

Traducción: Pau Fernández

Generación: 2026-03-12T16:22:23.323Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>