

## Assignment #4 – BINARY SEARCH TREES

### BST

Create a Binary Search Tree, which supports the following operations.

- **insert(tree, element)** – adds the node specified by element (which contains the data) into the BST specified by tree.
- **search(tree, key)** – searches for the data specified by key in the BST specified by tree.
- **findMin(tree)** – retrieves the smallest data in the BST specified by tree.
- **findMax(tree)** – retrieves the largest data in the BST specified by tree.
- **predecessor(tree, element)** – retrieves the inorder-predecessor of the node specified by element in the BST specified by tree.
- **successor(tree, element)** – retrieves the inorder-successor of the node specified by element in the BST specified by tree.
- **delete(tree, element)** – removes the node specified by element from the BST specified by tree.
- **inorder(tree)** – To do a recursive inorder traversal of the BST.
- **preorder(tree)** – To do a recursive preorder traversal of the BST.
- **postorder(tree)** – To do a recursive postorder traversal of the BST.

### Input - Output Format

The input consists of multiple lines, each one containing either one or two integers.

The first integer in the line can be 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 or 10 and each one has its own meaning:

- The integer 0 means stop the program.
- The integer 1 means, create a node which contains the next integer from the input as the data part, and then, insert this node into the BST. In this case, the next integer ( $\geq 0$ ) is given on the same line as the 1, separated by a space.
- The integer 2 means, search for the key specified by the next integer from the input, in the BST. In this case, the next integer ( $\geq 0$ ) is given on the same line as the 2, separated by a space. If the search is successful, output "FOUND". Otherwise, output "NOT FOUND".
- The integer 3 means find and output the minimum number in the BST. Print "NIL" if the BST is empty.

- The integer 4 means find and output the maximum number in the BST. Print “NIL” if the BST is empty.
- The integer 5 means find and output the inorder-predecessor of the data specified by the next integer from the input. In this case, the next integer ( $\geq 0$ ) is given on the same line as the 5, separated by a space. Output “NIL”, if the data exists in the tree, but there is no inorder-predecessor for the data. Output “NOT FOUND”, if the data is not present in the tree.
- The integer 6 means find and output the inorder-successor of the data specified by the next integer from the input. In this case, the next integer ( $\geq 0$ ) is given on the same line as the 6, separated by a space. Output “NIL”, if the data exists in the tree, but there is no inorder-successor for the data. Output “NOT FOUND”, if the data is not present in the tree.
- The integer 7 means delete the data specified by the next integer in the input from the BST, if present. In this case, the next integer ( $\geq 0$ ) is given on the same line as the 7, separated by a space. (Here, the data to be deleted is guaranteed to be present in the BST).

<u>Sample Input</u>	<u>Sample Output</u>
2 25	NOT FOUND
3	NIL
4	NIL
5 25 NOT FOUND	
6 25 NOT FOUND	
1 25	
2 25 FOUND	
3 25 4 25	
5 25 NIL	
6 25 NIL	
1 13 1 50 1 45 1 55 1 18	
2 10 NOT FOUND	
2 13	FOUND
2 35	NOT FOUND
2 55	FOUND
2 80	NOT FOUND
8 13 18 25 45 50 55 9	25
13 18 50 45 55	
10 18 13 45 55 50 25	
3 13 4 55	

5 13	NIL
6 13	18 5 18 13 6 18 25 5 25 18 6 25 45 5 45 25 6 45 50 5 50 45 6
50	55 5 55 50
6 55	NIL
7 55 7	13 7 25
3	18 4 50
5 18	NIL
6 18	45
5 45	18
6 45	50
5 50	45
6 50 7	NIL
45 7	
50 7	
18	
3	NIL
4 1	NIL
90	
3	90
4 0	90