16/May/24

lab - 3

Increasing order search tree.

Given the root of a binary search tree, rearrange the tree in-order so that leftmost node in tree is now the root of the tree, and every node has no left child and only one right child

```
Struct TreeNode * inorder_traversal (Struct TreeNode* root,
    root, Struct TreeNode ** new_root)
    {
      if (root)
        { root -> left = inorder_traversal (root ->
                                left, new_root);

      Printf ("Parent_root : %p", * new_root);
      (* new_root) -> right = root;
        * new_root = root;
      inorder_traversal (root -> right, new_root);
        return NULL;
        }
      return NULL;
    }
Struct TreeNode * IncreasingBST (Struct
                TreeNode * root)
    {
      Struct TreeNode * new_root =
        (Struct TreeNode*) malloc (sizeof (
```

```
struct Node));

new_root->val = INT_MIN;
    new_root->left = NULL;
    new_root->right = NULL;

struct TreeNode *Ptr = new_root;
struct TreeNode *return_root = new_root;

inOrder_traversal (root, &Ptr)
return return_root->right;
}
```

Case 1

Input: [5, 3, 6, 2, 4, Null, 8, 1, Null, Null, null, 7, 9]

Output: [1, null, 2, null, 3, null, 4, null, 5, null, 6 null, 7, null, 8, null, 9]

Case 2:

Input: [5, 1, 7]

Output: [1, null, 5, null, 7]

16/5/24