16/5/24

Leet code 3

⇒ Increasing Binary Search Tree

```
struct TreeNode * inorder_traversal (struct TreeN
        struct TreeNode ** new_root)
{
    if (root)
    {
        root -> left = inorder_traversal (root -> left,
        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 *) mal
                    (sizeof (struct TreeNode));
    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;
}
```

Output

Case 1

root = [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]

Expected = [1, null, 2, null, 3, null, 4, null, 5, null, 6, null, 7, null, 8, null, 9]

Case 2

root = [5, 1, 7]

output = [1, null, 5, null, 7]

Expected = [1, null, 5, null, 7]

16/5/24