

Min and Max In BST

By "inorder traversal" we can find minimum and maximum value in BST.

- left most bottom ele is the mini.
- right most bottom ele is the max.

Algorithm:

- S1: Create a BST
- S2: Sort BST in inorder method.
- S3: Visit each node in inorder traversal technique.
- S4: Left most bottom is minimum
- S5: Right most bottom is maximum.

Code ('C')

```
#include <stdio.h>
#include <limits.h>
#include <stdlib.h>

struct Node{
    int data
    struct Node* left, right;
};
```

```
int minValue(struct Node* root)
{
    if (root == NULL)
        return (-1);

    struct Node* curr = root;
    while (curr->left != NULL)
    {
        curr = curr->left;
    }

    return curr->data;
}
```

```
int maxValue(struct Node* root)
{
    if (root == NULL)
    {
        return (-1);
    }

    struct Node* curr = root;
    while (curr->right != NULL)
    {
        curr = curr->right;
    }

    return curr->data;
}
```

```
struct Node* createNode(int val)
{
    struct Node* node = (struct Node*) malloc
        (sizeof(struct Node));
    node->data = val;
```

```

node->left = node->right = Null;
return(node);
}

int main(){
    struct Node* root = createNode(5);
    root->left = createNode(4);
    root->right = createNode(6);
    root->left->left = createNode(3);
    root->right->right = createNode(7);
    root->left->left->left = createNode(1);
    printf("%d\n", minValue(root));
    printf("%d\n", maxValue(root));
    return 0;
}

```

→ Output

1
7

Complexity

Time = $O(h)$ → h - height
Space = $O(1)$

