**SOLUTION**

/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* TreeNode \*left;

\* TreeNode \*right;

\* TreeNode(int x) : val(x), left(NULL), right(NULL) {}

\* };

\*/

class Solution {

public:

TreeNode\* bstFromPreorder(vector<int>& preorder) {

return build\_tree(preorder,0,preorder.size()-1);

}

TreeNode\* build\_tree(vector<int>& preorder,int l,int r){

if(l>r)

return nullptr;

TreeNode\* root=new TreeNode(preorder[l]);

if(l==r)

return root;

int id=l+1;

while(id<=r && preorder[id]<preorder[l])

id++;

root->left=build\_tree(preorder,l+1,id-1);

root->right=build\_tree(preorder,id,r);

return root;

}

};

**TIME COMPLEXITY= O(N)**

**SPACE COMPLEXITY= O(N)**