TREE BASIC

#include <bits/stdc++.h>

using namespace std;

struct Node

{

int key;

struct Node \*left;

struct Node \*right;

Node(int k){

key=k;

left=right=NULL;

}

};

int main() {

Node \*root=new Node(10);

root->left=new Node(20);

root->right=new Node(30);

root->left->left=new Node(40);

}

Binary Search Tree – Iterative

#include <bits/stdc++.h>

using namespace std;

struct Node

{

int key;

struct Node \*left;

struct Node \*right;

Node(int k){

key=k;

left=right=NULL;

}

};

bool search(Node \*root, int x){

while(root!=NULL){

if(root->key==x)

return true;

else if(root->key<x)

root=root->right;

else

root=root->left;

}

return false;

}

int main() {

Node \*root=new Node(15);

root->left=new Node(5);

root->left->left=new Node(3);

root->right=new Node(20);

root->right->left=new Node(18);

root->right->left->left=new Node(16);

root->right->right=new Node(80);

int x=16;

if(search(root,x))

cout<<"Found";

else

cout<<"Not Found";

}

Binary Search – Recursive

#include <bits/stdc++.h>

using namespace std;

struct Node

{

int key;

struct Node \*left;

struct Node \*right;

Node(int k){

key=k;

left=right=NULL;

}

};

bool search(Node \*root, int x){

if(root==NULL)

return false;

if(root->key==x)

return true;

else if(root->key>x){

return search(root->left,x);

}else{

return search(root->right,x);

}

}

int main() {

Node \*root=new Node(15);

root->left=new Node(5);

root->left->left=new Node(3);

root->right=new Node(20);

root->right->left=new Node(18);

root->right->left->left=new Node(16);

root->right->right=new Node(80);

int x=16;

if(search(root,x))

cout<<"Found";

else

cout<<"Not Found";

}