COMSC-200
Lab 7

Ryan Jacoby

18 October 2020

1 Search Tree

```cpp
// Ryan Jacoby

#include<iostream>

using namespace std;

class Node {
private:
    Node * left;
    Node * right;
    int val;
public:
    Node(int);
    Node * search(int);
    Node * insert(int);
    void setLeft(Node *);
    void setRight(Node *);
    int getVal();
};

Node::Node(int val) {
    this->left = nullptr;
    this->right = nullptr;
    this->val = val;
}

Node * Node::search(int val) {
    if(this->val == val) return this;
    if(this->val > val) return this->left->search(val);
    return this->right->search(val);
}

Node * Node::insert(int val) {
    if(val < this->val) {
        if(this->left == nullptr){
            this->left == new Node(val);
            return this->left;
        }
        this->left->insert(val);
    }
    else if(val > this->val) {
        if(this->right == nullptr){
            this->right == new Node(val);
            return this->right;
        }
        this->right->insert(val);
    }

    return this;
}

void Node::setLeft(Node * n) {
    this->left = n;
}

void Node::setRight(Node * n) {
    this->right = n;
}

int Node::getVal() {
    return val;
}

int main() {
    Node head = Node(50);
    head.insert(30);
    head.insert(80);
    head.insert(10);
    head.insert(60);
    head.insert(70);

    cout << head.search(70)->getVal();
    return 0;
}
```

Listing 1: main.cpp