

# COMSC-200

## Lab 7

Ryan Jacoby

18 October 2020

### 1 Search Tree

```
1 // Ryan Jacoby
2
3 #include<iostream>
4
5 using namespace std;
6
7 class Node {
8 private:
9     Node * left;
10    Node * right;
11    int val;
12 public:
13    Node(int);
14    Node * search(int);
15    Node * insert(int);
16    void setLeft(Node *);
17    void setRight(Node *);
18    int getVal();
19 };
20
21 Node::Node(int val) {
22     this->left = nullptr;
23     this->right = nullptr;
24     this->val = val;
25 }
26
27 Node * Node::search(int val) {
28     if(this->val == val) return this;
29     if(this->val > val) return this->left->search(val);
30     return this->right->search(val);
31 }
32
33 Node * Node::insert(int val) {
34     if(val < this->val) {
35         if(this->left == nullptr){
36             this->left == new Node(val);
37             return this->left;
38         }
39         this->left->insert(val);
40     }
41     else if(val > this->val) {
```

```

42         if(this->right == nullptr){
43             this->right == new Node(val);
44             return this->right;
45         }
46         this->right->insert(val);
47     }
48
49     return this;
50 }
51
52 void Node::setLeft(Node * n) {
53     this->left = n;
54 }
55
56 void Node::setRight(Node * n) {
57     this->right = n;
58 }
59
60 int Node::getVal() {
61     return val;
62 }
63
64 int main() {
65     Node head = Node(50);
66     head.insert(30);
67     head.insert(80);
68     head.insert(10);
69     head.insert(60);
70     head.insert(70);
71
72     cout << head.search(70)->getVal();
73     return 0;
74 }

```

Listing 1: main.cpp