

```

1  //*****
2  // Project #3 Binary Search Tree
3  // Name: Ben Diekhoff
4  // Data Structures and Algorithms
5  // Date: 02/27/2019
6  //*****
7  /*
8  This program creates a binary search tree class. It adds nodes using
9  dynamically allocated memory from the heap. It takes input from cin through
10 "file.txt" and adds, removes, and returns the rank of nodes in the tree.
11 It also returns the amount of nodes within the range of a pair of integers.
12
13             COMPLEXITY
14 The complexity of this program is  $O(N)$ .
15 The most complex method in the program is  $O(N)$ . Many methods, like search,
16 insert, and remove would be  $O(\log N)$  on a perfectly balanced tree.
17 However, this tree is nearly flat, so these functions are all closer to
18  $O(N)$ . Additionally, other methods, like remove, numNodes, and inOrderPrint
19 need to access every node in the tree, so they are naturally  $O(N)$ .
20
21 //*****
22 I have written the entire program as turned in and have not copied this
23 code, or parts of this code from the internet or another student.
24 Signature_____
25 //*****/
26
27
28 // Source.cpp
29
30 #include <iostream>
31 #include <iomanip>
32 #include <initializer_list>
33 #include "BST.h"
34 using namespace std;
35
36 int main() {
37     BST T;
38     int N, M, K, addNum, delNum, mNum, a, b;
39
40     // Load numbers 1 - 1000
41     for (int i = 1; i <= 1000; i++) {
42         T.insert(i);
43         cout << setw(4) << i << " ";
44
45         if (i % 15 == 0)
46             cout << "\n";
47     }
48     cout << endl << endl;
49
50     //Insert and remove numbers from file.txt
51     cin >> N;
52     for (int i = 0; i < N; i++) {

```

```
53     cin >> addNum >> delNum;
54     T.insert(addNum);
55     T.remove(delNum);
56 }
57
58 // Print rank
59 cin >> M;
60 for (int i = 0; i < M; i++) {
61     cin >> mNum;
62     cout << T.rank(mNum) << endl;
63 }
64 cout << endl;
65
66 // Print range
67 cin >> K;
68 for (int i = 0; i < K; i++) {
69     cin >> a >> b;
70     cout << T.range(a, b) << endl;
71 }
72
73 return 0;
74 }
75
76
77
```