UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG
SCHOOL OF COMPUTER SCIENCE & APPLIED MATHEMATICS

COMS1017: Intro. to Data Structures & Algorithms
Lab Test 3

Prof. Richard Klein

10 October 2024

- This is a closed book test. You may not access any other websites, files, or accounts.
- Answer all the questions and submit your code to the submission on Moodle.
- You have 2.5 hours.
- There are 90 Marks available. **90 = 100%**
- Your code *must* compile. Code that does not compile will receive 0. Code will be compiled using `g++ -std=c++23 bst.cpp` on the marker. It will be run with `./a.out < testcase.txt`
- Presentation Errors are acceptable and will still get 100%. In all test cases that have multiple values, it is acceptable to print an extra space at the end of the list – the test cases on Moodle *will* have this additional space.
- Some seen test cases are provided in a separate folder called `io`. These test cases are on Moodle as a sanity check, but are not worth any marks.
- If you are using QtCreator, ensure that you tell it to "Run in terminal" by following the steps in the Figure 1.
- Ensure that the folder containing your code does not have brackets: `Downloads/handout(1)/`. This prevents QtCreator from being able to compile and it will show you a `makefile error`. Simply remove the brackets from the path, reopen the project, and QtCreator will work correctly.

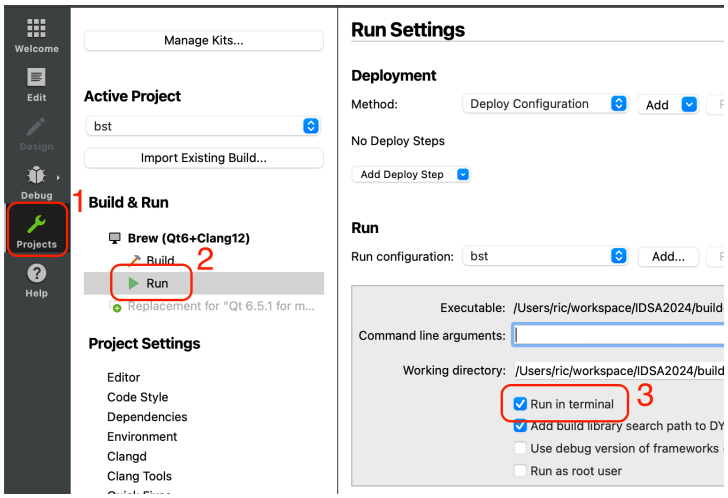

Figure 1

Instructions

You are given a Binary Search Tree (BST) skeleton on Moodle. You may use this skeleton to assist you or you may write your own code from scratch. The code provided includes a working `main` function that reads the trees correctly and calls the relevant methods. This code is correct and you do not need to change it.

This test is marked entirely using I/O marking. Your code should take as input a list of numbers separated by spaces until it reads a `-1`. These numbers should be inserted into a binary search tree in the order they appear. Once the tree is built, you should read a command. Based on this command, your code should call the relevant function on your tree. The provided skeleton code does this correctly.

Question 1 Traversals [30 Marks]

Implement the methods to print out the preorder, inorder and postorder traversals of the binary search tree. It should be called with the `pre`, `in`, and `post` commands. [30]

Sample In

```
42
80
25
10
75
99
15
-1
pre
```

Sample Out

```
42 25 10 15 80 75 99
```

Sample In

```
42
80
25
10
75
99
15
-1
in
```

Sample Out

```
10 15 25 42 75 80 99
```

Sample In

```
42
80
25
10
75
99
15
-1
post
```

Sample Out

```
15 10 25 75 99 80 42
```

Question 2 Descending Order [10 Marks]

Implement a method to print the values of the list in descending order. This function should be called with the `desc` command. [10]

Sample In

```
42
80
25
10
75
99
15
-1
desc
```

Sample Out

```
99 80 75 42 25 15 10
```

Question 3 Depth [10 Marks]

The *depth* of a node is the number of edges between that node and the root. Implement a method to print the depth of a given number in the tree. The method should be called with the `depth` command followed by a value. You may assume that the given value will always be in the tree. [10] .

Sample In

```
42
18
50
20
43
51
19
30
52
-1
depth 20
```

Sample Out

```
2
```

Sample In

```
42
18
50
20
43
51
19
30
52
-1
depth 30
```

Sample Out

```
3
```

Question 4 Internal Nodes [10 Marks]

An internal node is a node that has at least one child (i.e. a node that is not a leaf). Write a function that prints out the number of internal nodes in the tree. The method should be called with the `internal` command. [10]

Sample In

```
42
80
25
10
75
99
15
-1
internal
```

Sample Out

```
4
```

Sample In

```
42
18
50
20
43
51
-1
internal
```

Sample Out

```
3
```

Question 5 Equality [15 Marks]

Write a function that compares two BSTs and outputs whether they represent the same tree (i.e. the structure). The input will be first tree, terminated by -1 as before, followed by the `equals` command, followed by the second tree, terminated by -1 as well. The function should print a `true` or `false`. Two trees are equal if both the values *and* structure of the tree are the same. [15]

Sample In

```
10
5
15
13
18
-1
equals
10
15
18
13
5
-1
```

Sample Out

```
true
```

Sample In

```
10
5
15
13
18
-1
equals
10
13
15
5
18
-1
```

Sample Out

```
false
```

Question 6 Distance [15 Marks]

The distance between two nodes in a tree is the number of edges on the path from the one node to the other. Write a method that receives two integers and prints out the distance between the nodes containing those values. You are guaranteed that both of these values will exist in the tree and that they are unique. [15]

Hint: The path may or may not pass through the root, one node may or may not be an ancestor of the other.

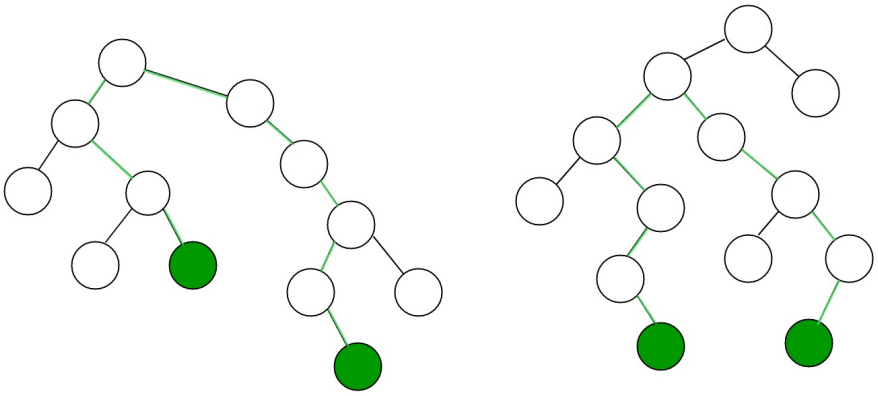

Figure 2: Nodes with a distance 8 from each other.

Sample In

```
20
10
5
15
12
17
22
24
30
25
29
32
-1
distance
29
17
```

Sample Out

```
8
```

Sample In

```
20
10
5
15
12
17
22
24
30
25
29
32
-1
distance
5
17
```

Sample Out

```
3
```

Sample In

```
30
10
38
1
12
0
5
3
4
14
13
19
15
-1
distance
4
15
```

Sample Out

```
8
```