

Data Structures HW3

Logistics

Due date: **5/21 (Sun) 23:59**

Submission

- Via LMS (**no email submission**)

- **Two files**

1. output.txt (answer file)

2. Zip file of your program (**Compress the files into one.**)

Note: **No restrictions on programming languages and platforms.**

Problem Description: DFS and BFS on BST

Write a program that 1) constructs a BST with a given set of integers, and 2) finds the sum of all integers up to the nth element of DFS/BFS. Assume that in all traversals, for each node, its left subtree is processed/visited before its right subtree. When doing DFS, apply a preorder.

A file "input.txt" is provided in the following format:

2

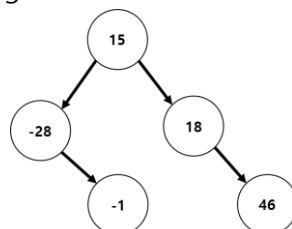
DFS: 3

15 18 46 -28 -1

BFS: 4

15 18 46 -28 -1

The number in the first line (i.e., 2) shows the number of test cases to be processed. From the second line, each test case is given over two lines. The second line "DFS: 3" tells that you need to apply DFS on the tree, and to add up all integers up to the 2nd element of the DFS. The numbers for this tree i.e., 15 18 46 -28 -1, are given in the next line. Note that when you construct a BST, **you must add nodes in the order in which they are written in the file.** That is, in this example, 15 should be added first into the tree, then 18, 46 and so on. As a result, the corresponding BST would be constructed as the following:



Applying preorder DFS to this tree, we have the traversal sequence of $15 \rightarrow -28 \rightarrow -1 \rightarrow 18 \rightarrow 46$, and thus the sum up to the third element, -1, would be -14 ($= 15 - 28 - 1$), which is the answer to the first case. Similarly, for the next case, the corresponding BFS sequence is $15 \rightarrow -28 \rightarrow 18 \rightarrow -1 \rightarrow 46$, and the sum up to the fourth number is 4 ($= 15 - 28 + 18 - 1$).

Finally, your "output.txt" should be as follows:

-14

4

Make sure you follow the output format (one answer per line). You are free to use existing queue and stack libraries for doing this homework; but you still have to implement the tree related features yourself. You can also use the following facts:

1. The maximum number of nodes is limited to 50.
2. All integers lie between -50 and 50.

Evaluation Policy (10 pts in total)

Score (10pts) = Accuracy (5pts) + Implementation (5pts)

Accuracy: $5 \times \# \text{ right answers} / \# \text{ total cases}$

Penalties

1. Unable to build code \rightarrow Implementation ≤ 2
2. **Plagiarism \rightarrow Score = -5 (will affect your overall grade)**
3. Late Submission \rightarrow Report $= 2$
4. **Wrong output format** and missing files (in case you forget to submit output.txt, or...)
 \rightarrow Accuracy ≤ 2