CS2705 : Programming and Data Structures

Assignment 2

September 19, 2022

Submission date: Oct 3, 2022 Max. Marks : 8

Instructions

- The assignment is graded. All questions are compulsory and have to be solved individually.
- You are required to submit the code on repl.it by following the given instructions.
- You are expected to write code completely on your own. Use of unfair means found will be penalized and reported to appropriate higher level committees.

1. (2 points) A mining site contains silver mines surrounded by a forest. The site is represented by a binary tree, that is, the nodes represent the sites and the links (left and right) represent the pathways between the sites. At each node, the value represents the amount of silver found in that site.

   A miner must follow the rules:

   - The silver can be mined at any site and accumulated but it can be claimed only at the leaf site.
   - In order to claim the silver at a site, that site must be mined.
   - If the silver at a site is mined then the silver at its parent site must be mined.
   - A miner must mine the root site.
   - From a given site, the miner can only take either the left or the right path (if possible).

   The miner has a desired amount that he wishes to claim from the site. The goal is to compute the number of paths for the miner to claim the exactly the desired value of silver to be claimed.

   Assume (without loss of generality) that the amount of silver is a positive integer.

   Input format. The first line contains the number of sites. The second line contains the level order traversal of a binary tree representation of the mining site. The level order traversal of the binary tree contains value = -1 if the site is missing, wherein the NULL (-1) nodes occurring towards the end are truncated. The third line contains a desired value V to be claimed.

   Output format. The number of ways in which the miner can claim the silver of value V

   Example.

   Input:

   8

10 5 20 10 20 15 5 -1 10

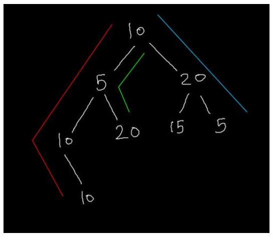
35

Output:

3

Explanation:

The given binary tree is pictorially represented as follows.



Starting from the root site, there are three pathways (shown in red, cyan and green) the miner can follow to claim the silver of value 35.

2. (3 points) You are given an array of integers that represents the scores of Virat Kohli in his recent matches. Your task is to figure out after each match, how many matches his fans had to wait before they saw him beating his last score. If a score was not beaten by a future match, then consider that as an infinite wait.

   Design and implement an algorithm with time complexity $O(n)$ where n is the size of the input array.

   You are free to use Standard Template Library for data structures if you choose to use them.

   Input format. The first line contains the value of n, that is, the size of the array. The second line contains the n scores themselves, each separated by a white-space.

   Output format. The array of size n, values separated by a white-space, where the i-th value represents the number of matches the fans had to wait after the i-th match to see Virat beating his score in the i-th match.

   Example 1.

   Input:

   8

   63 64 65 61 59 62 66 63

   Output:

   1 1 4 2 1 1 Inf Inf

   Example 2.

   Input:

   7

   50 48 0 13 90 1 108

   Output:

   4 3 1 1 2 1 Inf

3. (3 points) Given a binary tree of non-negative integers, you are required to modify it as follows. For every node h in the tree, if h is a right child node then replace the value in h by the sum of the values originally in the parent of h, the values in the left-subtree of the parent of h and h itself.

Input format. The first line contains the number of nodes in the tree. The second line contains the level order traversal of the binary tree where each value is separated by a white-space, and if a node is absent, its value is -1, wherein the NULL (-1) nodes occurring towards the end are truncated.

Output format. Level order traversal of the modified binary tree **without** the missing nodes.

Example.

Input:

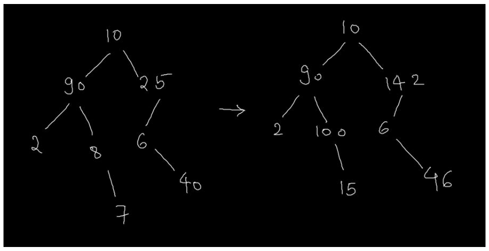
8

10 90 25 2 8 6 -1 -1 -1 -1 7 -1 40

Output:

10 90 142 2 100 6 15 46

Explanation: The given and the modified binary trees are represented pictorially as follows. For instance, the value 25 is replaced with sum of parent node's value (10), the values occurring in the left sub-tree of the parent node $(90 + 2 + 8 + 7)$ and the value in the node itself (25).


10
90
25
2
8
6
40
7
→
10
90
142
2
100
6
15
46