CS69201: Computing Lab-1
Take Home Assignment - AVL Tree
August 14, 2024

========== Instructions==========

1. In the case of user input assume only valid values will be passed as input.
2. You can use C or C++ as the programming language. **However, you are not allowed to use any STL libraries in C++**
3. Regarding Submission: For each question create a separate C file. -> <rollno>_Q1.c, <rollno>_Q2.c, <rollno>_Q3.c,<rollno>_Q4.c. Create a zip file of all these C files in the name <rollno>_A4_Takehome.zip and submit it to Moodle. For example, if your roll number is 24CS60R15, then your file name will be 24CS60R15_Q1.c, 24CS60R15_Q2.c, 24CS60R15_Q3.c, 24CS60R15_Q4.c and your zip file name will be 24CS60R15_A4_Takehome.zip.
4. **Inputs should be taken from a file and outputs should be printed to a file named output.txt. The input file name will be passed as a command line argument.**

==========

You can use the following numbers for each option in the menu:
1 = insert
2 = search
3 = delete
4 = height
5 = Preorder
6 = Postorder
7 = Inorder
8 = Level order
9 = Zigzag order
10 = Indicates the last line in the input

Question 1:

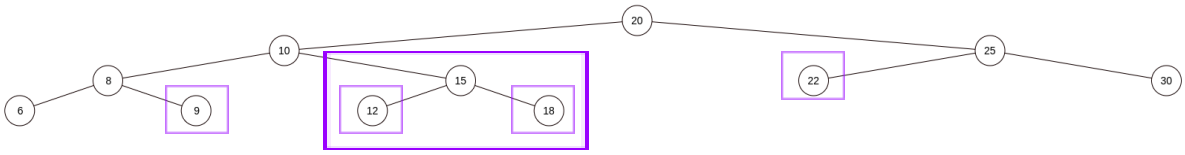
Construct an AVL tree from the user input values and find the number of subtrees that are present within the given range. We can tell a tree is within the range if its left subtree, right subtree, and the root is within the range low and high, inclusive.

Inputs should be in menu-driven format as shown in the example.
10 = last line along with the lower and upper bound value

Sample Input:
1 15
1 43
1 25
3 43
1 20
1 22
1 30
1 18
1 10
1 8
1 9
1 12
1 6
10 8 22

Sample Output:
No of subtrees within the range = 5

Explanation: Each box represents a subtree within the given range



Question 2:

We have a BST and need to burn the entire tree starting from a single node. Given a target node, determine the time to burn the entire tree. Assume that the fire starts at the target node at t=0, and in each time unit (t=1), it spreads to the parent node and all child nodes.
Print the nodes getting burned at each time.

Input should be menu-driven
10 = last line along with the target node

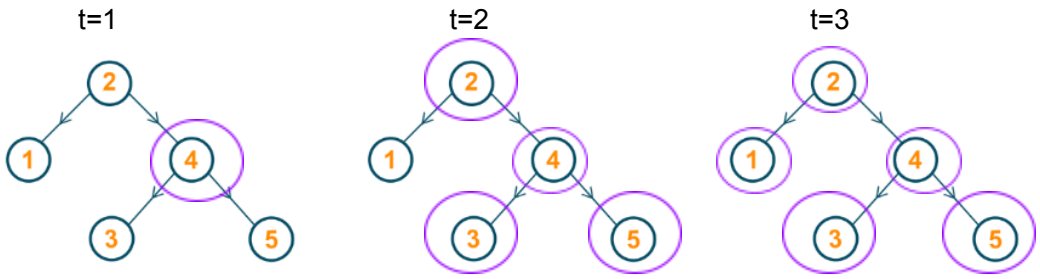
Sample Input:

1 2
1 1
1 4
1 3

1 5
10 4

Sample Output:
At time t=0, nodes burned = 4
At time t=1, nodes burned = 2,3,5
At time t=2, nodes burned = 1
Total time = 2

Explanation



Question 3:

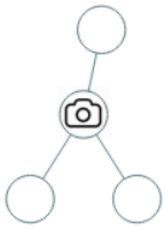
A teacher has arranged students into groups, each working in separate rooms. The rooms are connected in a structure resembling a BST, where each room corresponds to a node in the tree. The teacher wants to install cameras in some rooms, where each camera can monitor the room it's in, its connected parent room, and its immediate child rooms. Determine the minimum number of cameras needed to monitor all the rooms in the tree.

Sample Input:

1 10
1 5
1 3
1 6
10
Sample Output:

1

Explanation:



Question 4:

Given a binary tree print its

1. Vertical order
2. Right view
3. Left view
4. Top view
5. Bottom view
6. Boundary view

The inputs will contain the inorder and preorder of the binary tree. In the input file, the first one is the inorder and the second is the preorder.

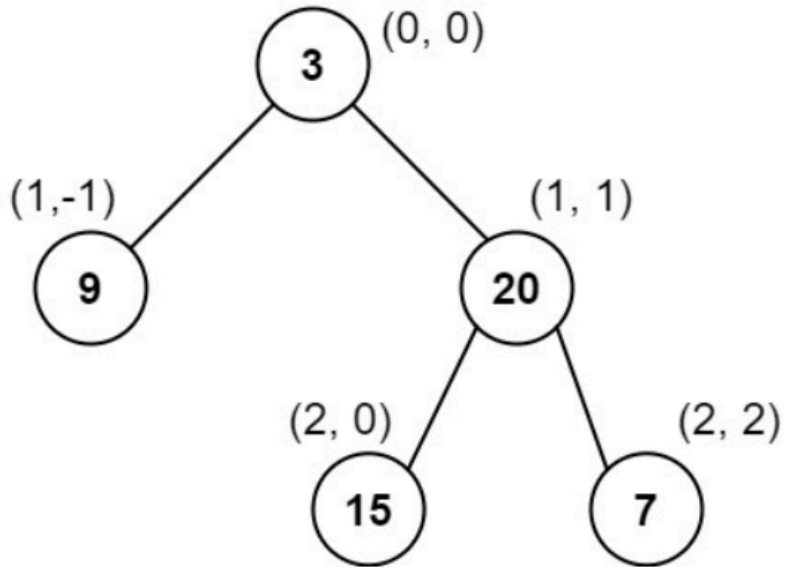
Sample Input:

5

9 3 15 20 7

3 9 20 15 7

Hint and Output:



For each node at position (row, col), its left and right children will be at positions (row + 1, col -1) and (row + 1, col + 1) respectively. The root of the tree is at (0, 0).

1. Vertical Order: It is a list of top-to-bottom orderings for each column index starting from the leftmost column and ending on the rightmost column
2. Right view: The nodes that are present on the rightmost side for each row
3. Left view: The nodes that are present on the leftmost side for each row
4. Top view: The nodes that are present in the topmost for each column
5. Bottom view: The nodes that are present in the bottommost of each column
6. Boundary view: The nodes that are present in the leftmost, bottom-most, right-most

For the above tree:

1. Vertical Order: [9] [3,15] [20] [7]
2. Right view: 3 20 7
3. Left view: 3 93 15
4. Top view: 9 3 20 7
5. Bottom view: 9 15 20 7
6. Boundary view: 3 9 15 7 20