DSA ASSIGNMENT 3

Deadline: 11:59 PM, 20/04/2025

Instructions:

1. **Do any one of the two assignments. If you choose to do both assignments, best of 3 will be evaluated from all the 4 assignments given so far.** ***Bonus Assignment 3B will be appended to this document soon.***

2. Assignments are to be attempted individually.

3. Submit assignment (single/both) as a single zipped folder (**A2_⟨RollNumber⟩.zip**) containing a pdf file for all the theory questions (**must also contain the explanation of programming questions and screenshots of relevant test cases**) and "C" files (code_⟨QuesNum⟩_⟨RollNumber⟩.c) for programming questions, for example "code_3_1_⟨RollNumber⟩.c" for programming question 3.1.

4. Please read the instructions given in the questions carefully. In case of any ambiguity, post your queries on Google Classroom at least 3 days before the deadline. **No TA will be responsible for responding to the queries after this.**

5. All the TAs will strictly follow the rubric provided to them. **No requests will be entertained related to the scoring strategy.**

6. **The use of generative tools (such as ChatGPT, Gemini, etc.) is strictly prohibited.** Failure to comply may result in severe consequences related to plagiarism.

7. **Extension and Penalty clause:**

   - Even a 1-minute late submission on Google Classroom will be considered late. Please turn in your submissions at least 5 minutes before the deadline to avoid any hassle.
   - Not explaining the answers properly will lead to zero marks.

Assignment 3A [70 Marks]

All the questions need to be attempted.

1. BST Traversal (15 marks)

A Binary Search Tree (BST) contains the following integer keys:

30, 15, 60, 7, 22, 45, 75, 17, 27.

1.1 Draw the BST formed by inserting these elements in the given order. (5 marks)
1.2 List the in-order, pre-order, and post-order traversals of the tree. (5 marks)
1.3 Suppose you now delete the node with key 15. Show the updated BST and explain the deletion process briefly. (5 marks)

2. Priority Queue (15 marks)

A hospital maintains a priority queue of patients based on their condition severity (lower value = higher priority). The following operations occur:

1. **enqueue("Alice", 4)**
2. **enqueue("Bob", 2)**
3. **enqueue("Carol", 5)**
4. **enqueue("Dave", 1)**
5. **dequeue()**
6. **enqueue("Eve", 3)**
7. **dequeue()**

2.1 After all operations, list the names in the priority queue in their current order (as per the priority). (7.5 marks)
2.2 Explain how the underlying data structure (like a binary heap) ensures correct order in O(log n) time for both `insert` and `delete`. (7.5 marks)

3. Search Tree

The Legacy of the Tree Kingdom (20 Marks)

In the ancient Kingdom of Arboris, a mystical tree stood at the center of the realm — the Tree of Fortune. Each branch bore a golden number, representing the wealth of a noble house in the kingdom. The tree was structured in a very particular way: every noble house on the left of a branch held less wealth, and those on the right held more — a clear hierarchy rooted in centuries of tradition.

But times were changing. The wise queen of Arboris declared a new law: from now on, each house's wealth must reflect not only its own fortune but also the accumulated wealth of all mightier houses.

In simpler terms, every house was to inherit the collective riches of all houses more powerful than itself as a symbol of unity and shared prosperity.

The royal scholars were tasked with updating the Tree of Fortune accordingly. The challenge? Traverse the tree in just the right order to redistribute the wealth — no house left behind and no riches miscounted.

You are given the root of the Tree of Fortune, which is a binary tree representing a Binary Search Tree (BST). Your task is to transform this tree into a Prosperity Tree, where each node's value becomes the original value plus the sum of all values strictly greater than it in the tree.

Return the updated root of the tree after completing the transformation.
Example Input: 5 2 13
Example Output: 18 20 13

- Implement the above code in C/CPP (10 Marks).
- How does this transformation affect the original BST properties (5 Marks)?

- What is the time and space complexity of your approach? Can it be optimized (5 Marks)?

4. Heap

The Skyline Watchtower Mission (20 Marks)

In the towering city of Skyreach, each building serves as a watchtower to keep an eye on the horizon. These towers are lined up along the main boulevard, and each one has a specific height.

To improve surveillance efficiency, the city's engineers have developed an advanced hover drone that patrols over the city. The drone can monitor exactly k towers at a time, capturing intelligence from the tallest tower in its current view.

As the drone glides from left to right across the skyline, one tower at a time, you are tasked with reporting the height of the tallest tower visible to the drone at each of its positions.

Given an array of heights[], where each element represents the height of a watchtower, and an integer k represents the number of towers the drone observes at once, return an array of integers where each element is the maximum height the drone sees in that window as it moves across the city.

Example:
Input: heights = [1, 3, 2, 4, 15, 3, 10, 7] k = 3
Output: [3, 4, 15, 15, 15, 10]

- Implement the above approach using Heap (13 Marks).
- Briefly explain if there is any other more efficient approach to solve the above problem. Also, provide the pseudo-code for the same (7 Marks).