ESO207A: Data Structures and Algorithms

Theoretical Assignment 1

Due Date: 3rd September 11:59pm, 2025

Total Number of Pages: 3 Total Points 140

Instructions-

1. For submission typeset the solution to each problem and compile them in a single pdf file. Hand-written solutions will not be accepted. You can use LaTeX or Word for typesetting.
2. Start each problem from a new page. Write down your Name, Roll number and problem number clearly for each problem.
3. For each question, give the pseudo-code of the algorithm with a clear description of the algorithm. Unclear description will receive less marks. Less optimal solutions will receive only partial marks.
4. Assume that sorting would have $O(n \log n)$ complexity.

Question 1. Maximal Product sum

You are given two integer sequences:

$$a = [a_1, a_2, \ldots, a_n]$$
$$b = [b_1, b_2, \ldots, b_n]$$

Each sequence contains n elements.

We define:

$$P = \max_{\pi} \sum_{i=1}^{n} a_i \cdot b_{\pi(i)}$$

Here, π is any permutation of the set $\{1, 2, \ldots, n\}$.

In other words, P represents the **maximum possible dot product** between sequence a and any rearrangement of sequence b. This rearrangement is conceptual only—no actual reordering of b needs to be performed.

(a) (5 points) Design an algorithm to determine the value of P and write a neat Pseudo code for it.

(b) (15 points) Now suppose over the time, the sequences a and b are updated through q operations. Each operation is described by two integers (o, x) where:

- $1 \leq x \leq n$
- If $o = 1$, increment a_x by 1.
- If $o = 2$, increment b_x by 1.

Your task is to design an algorithm to find out the value of P in q different situations. Provide a neat Pseudocode for the problem.

(c) (10 points) Give proof of correctness and time complexity analysis of your approach for part (b).

Question 2. **Game Score Maximization**

You are given an array nums of positive integers, of size n. You have to play a game where you have to maximize the score of a game. Initially your score is 1. Now you will be given k coins using which you will have to increase your score.

With each coin you can do the following operation:

1. Choose $[L, R]$ which you have not chosen previously.
2. Choose an element x in the range $[L, R]$ which has the highest number of prime factors. If more than one element has equal factors, take the element with the smallest index as x.
3. Now multiply x with the score.

(a) (20 points) Design an algorithm for determining the final score of the game

(b) (10 points) Give proof of correctness and time complexity analysis of your approach for part (a).

Question 3. **Wealth accumulate**

You are given the root of a family tree, represented as a binary tree, where each person (node) except the latest generation has two children: a left child and a right child. Each node contains the initial bank balance of that person and pointers to its children and there are total n nodes

At the end of each year, the following events occur simultaneously for all persons:

1. **Wealth Growth:** The wealth of each person is quadrupled due to bank interest.
2. **Donation:** Each person donates half of their wealth to exactly one of their children.
   - In the first year, the donation goes to the left child.
   - In the second year, the donation goes to the right child.
   - The process alternates between left and right in subsequent years.

Note: The amount of donation for any person each time is fixed for that year before receiving donation from others.
Given the initial bank balances, determine the wealth of each person after k years.

(a) (40 points) Design an algorithm for determining the wealth of each person after k years, where $k >> n$

(b) (10 points) Give time complexity analysis of your approach for part (a).

Question 4. **The King's Punishment**

King Robert Baratheon wished to inspect all his knights, so he ordered them to stand in a single line. Each knight has a certain height, and it is guaranteed that no two knights have the same height. The King, who is particular about order, declared a punishment rule for disorderly knights:

- A knight is punished once for every taller knight standing before him in the lineup.

Your task is to determine, for each knight, the total number of punishments he receives. Devise an algorithm that reports this array in $O(n \log n)$ time.

(a) (5 points) Describe a brute force algorithm to compute the punishments received by each knight. What would be its time complexity?

(b) (16 points) Suppose we insert the knights one by one in the order they stand. Can we maintain their heights in a data structure that allows us to quickly compute the punishments for the newly added knight? Describe the required operations of such a data structure and write the pseudocode for them.

(c) (3 points) Do you need to do something additional to ensure these operations remain efficient as the number of knights grows? (Think about what happens if the data structure becomes unbalanced.)

(d) (6 points) Analyze the time complexity of both the update and query operations in your data structure. Hence, give the overall time complexity of your algorithm.