DSA Written Assignment 4

Deadline: April 2nd

Q1: Weighted Median

You have n items, each associated with an integer *value* and a positive *weight*. Let S be the set of these items (all items will have distinct *weights* and *values*). For any subset $A \subseteq S$, define $w(A)$ to be the sum of the weights of the items in A. For an item $x \in S$, let $S_{<x}$ denote the set of items whose values are **strictly less** than the value of x, and let $S_{>x}$ denote the set of items whose values are **strictly greater** than the value of x.

A **weighted median** is any item $x \in S$ such that:

$$w(S_{<x}) \leq \frac{w(S)}{2} \quad \text{and} \quad w(S_{>x}) \leq \frac{w(S)}{2}.$$

Given two unsorted arrays $V[1..n]$ and $W[1..n]$, where $V[i]$ is the value of the i-th item and $W[i]$ is its weight, describe and analyze an algorithm to compute a weighted median in $O(n)$ time.

Note that $S_{<x}$ and $S_{>x}$ are just definitions and only arrays V and W are given to you.

Q2: Finding the Median of Five Distinct Numbers Using a Decision Tree

You have an array $A[1..5]$ containing five distinct numerical values. Your task is to determine the median of these values (the element that ranks third in sorted order) using at most 6 comparisons. Rather than providing pseudocode, you must illustrate your solution using a *decision tree*. A decision tree is a binary tree where each internal node represents a comparison of the form:

$$A[i] \geq A[j] \ ?$$

and each leaf node identifies the position of the median element in the array (e.g., i if $A[i]$ is the median).

Construct such a decision tree, ensuring that the total number of comparisons from the root to any leaf does not exceed 6. Explain why your tree correctly identifies the median in all possible orderings of the array's elements.

Q3: Identifying Top Programmers in a Hackathon

Imagine a hackathon with 100 talented programmers. Due to hardware limitations, only 10 computers are operational at any given time, so only 10 programmers can compete simultaneously in one session. In each session, the participating programmers are ranked from 1^{st} to 10^{th} based solely on their performance (with no ties). Furthermore, the programmers' skills are perfectly transitive; that is, if programmer A outperforms programmer B and programmer B outperforms programmer C, then A will always outperform C.

Your tasks are as follows:

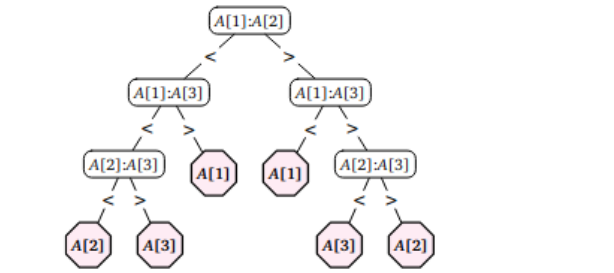

Figure 1: Example of a decision tree for finding the median of a 3-element array using at most 3 comparisons.

(a) Devise a strategy that determines the best programmer among the 100 by conducting at most **11** sessions.

(b) Prove that it is impossible to guarantee the identification of the best programmer in only **10** sessions.

(c) Inspired by the presence of 2 legendary coders from previous competitions, 1 promising newcomer, and 1 expert mentor among the audience, the hackathon organizers decide to host an extra session. This session aims to determine not only the best programmer but also the 2^{nd}, 3^{rd}, and 4^{th} best programmers. Explain how to identify the top four programmers by using a total of at most **12** sessions.

Submission

- Submit a single pdf named roll_number .pdf in Moodle. For example, 2020111016.pdf
- The submission can be handwritten or typed. Illegible handwriting will be given zero.
- State your assumptions clearly.