Algorithms: CSE 202 — Homework 1

For each problem, provide a high-level description of your algorithm. Please make sure to include the necessary details that are crucial for its correctness and efficiency. Prove its correctness and analyze its time complexity.

Problem 1: Maximum weight subtree

The maximum weight subtree is as follows. You are given a tree T together with (not necessarily positive) weights $w(i)$ for each node $i \in T$. A subtree of T is any connected subgraph of T, (so a subtree is not necessarily the entire subtree rooted at a node). You wish to find a subtree of T that maximizes $\sum_{i \in S} w(i)$. Design an efficient algorithm for solving this problem. Note that there is a linear (in the number of nodes of the tree) time algorithm for this problem. You can assume that for each node in the tree T, you are given a list of its children as well as the parent pointer (except for the root node).

$O(n)$

Problem 2: Sorted matrix search

Given an $m \times n$ matrix in which each row and column is sorted in ascending order, design an algorithm to find an element. $O(m+n)$

Problem 3: Largest set of indices within a given distance

You are given a sequence of numbers $a_1, \ldots, a_n$ in an array. You are also given a number k. Design an efficient algorithm to determine the size of the largest subset $L \subseteq \{1, 2, \cdots, n\}$ of indices such that for all $i, j \in L$ the difference between a_i and a_j is less than or equal to k. There is an $O(n \log n)$ algorithm for this problem.

For example, consider the sequence of numbers $a_1 = 7, a_2 = 3, a_3 = 10, a_4 = 7, a_5 = 8, a_6 = 7, a_7 = 1, a_8 = 15, a_9 = 8$ and let $k = 3$. $L = \{1, 3, 4, 5, 6, 9\}$ is the largest such set of indices. Its size is 6.

Problem 4: Toeplitz matrices

A *Toeplitz matrix* is an $n \times n$ matrix $A = (a_{ij})$ such that $a_{ij} = a_{i-1,j-1}$ for $i = 2, 3, \ldots, n$ and $j = 2, 3, \ldots, n$.

1. Is the sum of two Toeplitz matrices necessarily Toeplitz? What about the product?
2. Describe how to represent a Toeplitz matrix so that two $n \times n$ Toeplitz matrices can be added in $O(n)$ time. use FFT
3. Give an $O(n \lg n)$-time algorithm for multiplying an $n \times n$ Toeplitz matrix by a vector of length n. Use your representation from part (b).
4. Give an efficient algorithm for multiplying two $n \times n$ Toeplitz matrices. Analyze its running time.

$O(n^2)$