(CS1.301) Algorithm Analysis and Design (Monsoon 2024) Date: 21.09.2024

Mid-Semester Examination

Alloted time: 90 minutes Total marks: 50

Instructions:

- There are a total of 5 questions with varying credit printed on pages 1 and 2.
- Discussions amongst the students are not allowed. No electronic devices (including smart watches) nor notes/books of any kind are allowed.
- Any dishonesty shall be penalized heavily.
- Any theorem/lemma/claim/fact that was proved in the class can be used without proof in the exam, only by *explicitly* writing its statement, and a clear remark that it was chosen from the class notes.
- Questions have been framed to be disambiguous, and queries will not be answered during the examination. In case you find any ambuguity, please mention that in your answer scripts and work with it. Answers got by misreading of questions may not be given credit.
- Be clear in your arguments. Partial marking is available for every question but vague arguments shall not be given any credit.
- Analysis of running times, and proofs of correctness need to be done unless explicitly asked not to.

Question 1

[2 + 4 + 2 marks]

1. Compute third primitive root of unity.
2. Construct the DFT and inverse DFT matrices of order 3×3.
3. Compute the DFT of the vector $(-1, 1, 1)$.

Question 2

[4 + 6 marks]

A directed graph $G = (V, E)$ is strongly connected if and only if every pair of vertices is strongly connected.

(a) Give an algorithm that computes all strong components of a graph in time at most $O(|V| \cdot |E|)$.

(b) Can this be improved to $O(|V| + |E|)$ by considering $G_{reverse}$ (which is the graph obtained by reversing edge directions)?

Question 3 [8 marks]

Suppose you are given a connected graph G, with edge costs that are all distinct. Prove that G has a unique minimum spanning tree.

Question 4 [4 + 6 marks]

Let $G = (V, E)$ be a connected graph on n vertices and m edges such that their edge weights are all distinct. Algorithm 1 presents a different algorithm than what we studied in our classes. Your task is to

(a) analyse the running time, and

(b) prove the correctness of this algorithm.

Algorithm 1: Boruvka's algorithm for Minimum Spanning Tree

Input: A weighted connected graph $G = (V, E)$ with unique edge weights.
Output: The minimum spanning tree T for G.

Let T be a subgraph of G initially containing just the vertices in V (with no edges, just isolated vertices);
while T *has fewer than* $|V| - 1$ *edges* **do**
  for *each connected component* C_i *of* T **do**
    Let $e = (u, v)$ be the smallest-weight edge in E with $u \in C_i$ and $v \notin C_i$;
    Add e to T (unless e is already in T);
return T

Question 5 [8 + 6 marks]

Recall the problem of finding the number of inversions. As in the text, we are given a sequence of n numbers $a_1, \ldots, a_n$, which we assume are all distinct, and we define an inversion to be a pair $i < j$ such that $a_i > a_j$.

1. We motivated the problem of counting inversions as a good measure of how different two orderings are. However, one might feel that this measure is too sensitive. Let's call a pair a **significant inversion** if $i < j$ and $a_i > 2a_j$.

   Give an algorithm to count the number of **significant inversions** between two orderings.

2. Consider a further generalization. In addition to the sequence $a_1, \ldots, a_n$, we are given weights $w_1, \ldots, w_n$ where $w_i \geqslant 1$ for each i. Now call a pair $i < j$ a **generalized significant inversion** if $a_i > w_j a_j$. Describe an algorithm that counts the number of **generalized significant inversions** in time $O(n \log^2 n)$ or better.