Problem set 3

Date : Jan 20, 2022 Algorithms-S22

Instructions

- The problem sets are not for submission and will not be graded. However, you are strongly encouraged to spend some time thinking about the questions. This might be helpful in internalizing some of the things that we would discuss in the lectures.
- To get the most out of problem sets, you are strongly encouraged to think about the problems on your own before discussing with others, consulting any references or looking at hints (that some of the problems might have).

Problems

1. Recall the Vandermonde matrices that we saw in the class. These matrices are defined based on n complex numbers $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ and for $i, j \in \{0, 1, \ldots, n-1\}$, the (i, j) entry of the matrix is α_i^j.

   Prove that such a matrix has full rank if and only if $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ are all distinct.

2. Let n be a power of 2 and let $\omega_n = e^{i \cdot 2\pi/n}$. Recall the DFT matrix D_n of order n, with its rows and columns being indexed by integers in $\{0, 1 \ldots, n-1\}$ and the (i, j) entry being equal to $\omega_n^{i \cdot j}$. Prove that D_n can be written as a product of $O(\log n)$ matrices, each with a constant number of non-zero entries in every row.

   Use this decomposition to conclude that the discrete Fourier transform of a vector can be computed in $O(n \log n)$ operations.

3. Let $G = (V, E)$ be a graph with n vertices in which each pair of vertices is joined by an edge. There is a positive weight $w_{i,j}$ for each edge (i, j) and the edges are assumed to satisfy the triangle inequality, i.e, $w_{i,k} \leq w_{i,j} + w_{j,k}$. For a subset V' of vertices, $G[V']$ denotes the (with edge weights as given above) induced on the vertices in V'.

   We have a set X of k terminals that must be connected by edges. A Steiner tree on X is a set Z of vertices with $X \subseteq Z \subseteq V$, together with a spanning tree T of the induced subgraph $G[Z]$. Weight of the Steiner tree is the weight of T.

   So that there is a function $f : \mathbb{N} \to \mathbb{N}$ and a constant $c \in \mathbb{N}$ such that the problem of minimum weight spanning tree on X can be solved in time $O(f(k) \cdot n^c)$.

4. Suppose you are given a directed graph $G = (V, E)$ with costs (possibly negative) on the edges c_e for $e \in E$ and a sink t. Assume that you also have finite values $d(v)$ for every $v \in V$. Someone claims that for every node $v \in V$, the quantity $d(v)$ is the cost of the minimum-cost path from node v to the sink t.

   (a) Give an $O(|E|)$ time algorithm to verify whether this claim is correct.

   (b) Assume that the distances are correct and $d(v)$ is finite for all $v \in V$. Now, the goal is to compute distances to a different sink t'. Give an $O(m \log n)$ algorithm for computing the distances $d'(v)$ for all nodes $v \in V$ to the sink t'.

5. Suppose we have a directed acyclic graph with costs on edges and the costs may be positive or negative but every cycle in the graph has a strictly positive cost. We are also given two vertices u, v in the graph. Design an algorithm that computes the number of shortest $u - v$ paths in the graph.

6. Design and analyze an algorithm that takes as input an undirected graph and determines whether $G = (V, E)$ contains a simple cycle of length four in time at most $O(|V|^3)$.

   You can assume that the input is given as an adjacency matrix or an adjacency list, whatever you find helpful.