ADSA
Problem Set 4

1. Give an $O(n^2)$ time algorithm to compute the length of a longest increasing subsequence in a given sequence of integers. A subsequence is an increasing subsequence if elements are sorted in an ascending order.

2. Give an $O(n)$ time algorithm to find a contiguous subarray with the largest sum in an array of integers. Casually describe the working of your algorithm.

3. Suppose you are given three strings of characters: X, Y, and Z, where $|X| = n$, $|Y| = m$, and $|Z| = n + m$. Z is said to be a shuffle of X and Y iff Z can be formed by interleaving the characters from X and Y in a way that maintains the left-to-right ordering of the characters from each string. For instance, “cchocohilaptes” is a shuffle of “chocolate” and “chips”, but “chocochilatspe” is not.

Give an $O(mn)$-time dynamic programming algorithm that determines whether Z is a shuffle of X and Y.

4. You are given a directed acyclic graph $G = (V, E)$ with real-valued weights and two distinguished vertices s and t. The weight of a path is the sum of the weights of the edges in the path. Give a dynamic-programming algorithm for finding a longest weighted simple path from s to t. What is the running time of your algorithm? Explain the working of your algorithm.

5. Is the path between two vertices in a minimum spanning tree necessarily a shortest path between the two vertices in the original graph? Give a proof or a counterexample.

6. Assume that all edges in the graph have distinct edge weights (i.e., no pair of edges have the same weight). Is the minimum spanning tree of this graph unique? Give a proof or a counterexample.

7. Suppose we are given the minimum spanning tree T of a given graph G (with n vertices and m edges) and a new edge $e = \{u, v\}$ of weight w that we will add to G. Give an $O(n)$ time algorithm to find the minimum spanning tree of the graph $G + e$.

8. Give an $O(|E| \log |V|)$ time algorithm to find an MST of an undirected, weighted, and connected graph G that contains a specific edge $e \in E(G)$.

Solutions

Solution 1

Read this.

Solution 2

Read this.

Solution 3

Read this.

Solution 4

Read this.

Solution 5

No. Consider a complete graph of three vertices where every edge has weight 1. Now it is easy to find counter-example in such a graph.

Solution 6

Read lemma 7.1 here.

Solution 7

Adding edge e to the MST creates a unique cycle. Delete the maximum weight edge on this cycle. Proving that what we get after doing this is indeed an MST is left to you.

Solution 8

First include e in the MST and then run Kruskal's algorithm. The correctness follows from the cut based lemmas we have seen for MSTs.