CS 4349.501 Homework 7

Due Wednesday October 25th, in class

October 18, 2017

Please answer the following questions. Each student must write their solutions in their own words and submit their solutions on paper at the beginning of class. ***Include your name and/or Net ID at the top of each page.*** And please staple your papers together if you can.

In order to give you a better opportunity to demonstrate your strengths than the midterm exam provided, this homework has only one regular problem and two extra credit problems. Each extra credit problem is worth 1/8 of the midterm. ***You are expected not to use any sources outside the course lecture notes, the textbook, or Jeff Erickson's notes while answering these questions, including other people. Part of your score will depend on you proving your algorithm is correct.*** You may want to look at the solutions for the midterm exam to see what level of detail I expect in these proofs.

1. (a) Describe and analyze an algorithm to compute the *maximum*-weight spanning tree of a given edge-weighted graph.

   (b) A *feedback edge set* of an undirected graph G is a subset F of the edges such that every cycle in G contains at least one edge in F. In other words, removing every edge in F makes the graph G acyclic. Describe and analyze a fast algorithm to compute the minimum weight feedback edge set of a given edge-weighted graph.

   The running time for both parts should be in terms of V and E, the number of vertices and edges in the input graph.

2. ***Extra credit*** (worth 1/8 a midterm): Suppose you are given an integer k and an array $A[1..n]$ of n *distinct* integers, sorted in increasing order. Describe and analyze a recursive algorithm to determine whether there is an index i such that $A[i] = i + k$ in $O(\log n)$ time. Slower but correct algorithms are worth a small amount of partial credit.

3. ***Extra credit*** (worth 1/8 a midterm): A *common supersequence* of two sequences/arrays $A[1..m]$ and $B[1..n]$ is another sequence that includes both the elements of A and the elements of B in order. For example, if the two sequences are `BIOLOGICAL` and `DIPLOMATICALLY`, a common supersequence of those two sequences is `DBIOPLOMATG-ICALLY`.

   For all integers i, j such that $0 \leq i \leq m$ and $0 \leq j \leq n$, let $SCS(i, j)$ denote the length of the *shortest* common supersequence of sequences $A[1..i]$ and $B[1..j]$.

   (a) Give a recurrence definition or describe a simple recursive algorithm for computing $SCS(i, j)$. You *do not* need to analyze the algorithm if you choose to describe one. Don't forget to explain why your solution is correct.

   (b) Use your solution from part (a) to describe and analyze an $O(n^2)$ time dynamic programming algorithm to compute the length of the shortest common supersequence of $A[1..m]$ and $B[1..n]$. ***Your solution to part (a) must be correct to receive any credit for this part.***