CSC373S **Problem Set 1** **Spring, 2013**

Due: February 1, beginning of lecture

NOTE: Each problem set only counts 5% of your mark, but it is important to do your own work (but see below). Similar questions will appear on the first term test. You may consult with others concerning the general approach for solving problems on assignments, but you must write up all solutions entirely on your own. Anything else is *plagiarism*, and is subject to the University's Code of Behavior. You will receive 1/5 points for any (non bonus) question/subquestion for which you say "I do not know how to answer this question". You will receive .5/5 points if you just leave the question blank.
Advice: Do NOT spend an excessive amount of time on any question and especially not on a bonus question. If you wish to spend "free time" thinking about (say) bonus questions that is fine but you should not sacrifice time needed for other courses.

1. (10 points)

   Suppose we change the definition of *compatible* so that now intervals are closed at both ends; that is, $[s_1, f_1]$ and $[s_2, f_2]$ are not compatible if $f_1 = s_2$. Determine whether or not the EFT greedy algorithm is an optimal algorithm for interval scheduling given the new definition of compatible. Specifically, give a counter-example if it is not an optimal algorithm or argue why the inductive proof (or promising partial solutions) of the existing proofs is still a correct proof of optimality.

2. (10 points)

   Modify the charging argument given for the ISP problem to show that the same EFT greedy algorithm is a 2-approximation algorithm for the JISP problem. Namely, show that there is a 2-1 function that maps the set of intervals in an arbitrary (say optimal) solution into the set of intervals constructed by the EFT algorithm.

3. (15 points) The following quesitons refer to the MST problem

   - Question 5.1(a) in DPV text
   - Question 5.6 in DPV text.
   - Give an example of a weighted graph where edge weights are not all distinct but still there is a unique MST.

4. (10 points)

   Question 5.14 in the DPV text.

5. (15 points) We wish redefine the *cost of a path* in various ways and then see if Dijkstra's shortest path algorithm will still optimally solve the least cost paths problem. For each of the following definitions of the cost of a path, state and justify whether or not Dijkstra's algorithm optimally solves the least cost problem. We assume a non negative cost $c(e)$ for each edge e in the graph. If Dijkstra's algorithm is not optimal then show a counter example. If Dijkstra's algorithm is still optimal then say what is the key observation in the proof that still holds.

   (a) (5 points) The cost of a path π is $\max_{e \in \pi} c(e)$

   (b) (5 points) The cost of a path π is $\min_{e \in \pi} c(e)$

   (c) (5 points) The cost of a path π is the average cost of an edge in π; that is, $cost(\pi) = \frac{\sum_{e \in \pi} c(e)}{|e : e \in \pi|}$

6. (15 points) Consider the following variant of the knapsack problem. There are n items and each item can be taken j times for $j \in \{0, 1, 3\}$ times. Assume further that the size of each item is an integer $\leq n^2$. Provide a dynamic programming (DP) algorithm for this problem.

   - (5 points) Provide a semantic array definition for computing the cost of an optimal solution.
   - (5 points) Provide a recursive (computational) definition for computing values of this array and briefly justify why your computational definition is equivalent to the semantic definition.
   - (5 points) What is the asymptotic complexity of your algorithm in terms of the number of arithmetic operations and comparisons as a function of n.

7. Problem 6.27 in the DPV text.