Due: Thursday, October 6, beginning of tutorial

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 which will cover material relating to both assignment 1 and assignment 2. 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. Depending on the enrollment and as an experiment, we may sometimes allow students to work in pairs and then submit one assignment. For problem set 1, you may work in pairs *only* for the bonus questions. Anything else is *plagiarism*, and is subject to the University's Code of Behavior. You will receive 1/5 points for any 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.

1. (a) Problem 7 of chapter 4 (pages 191,192). You must describe a greedy algorithm and prove that your greedy algorithm always provides an optimal schedule. Hint: use an exchange argument to establish optimality. [15 points]

   (b) Bonus question: Suppose that the final f_i processing time for each job had to be performed on another supercomputer (instead of a PC) and at most one job can use this supercomouter at a time. Describe a greedy algorithm and prove that your greedy algorithm always provides an optimal schedule. [20 points]

2. This problem concerns Kruskal's greedy algorithm for the MST problem. Let $G = (V, E)$ be a connected graph and $c : E \to \Re$ a cost function on the edges.

   (a) Would Kruskals algorithm still be an optimal algorithm if the cost function allowed negative cost edges? Hint: does the proof of optimality depend on the edge costs being positive?

   (b) Suppose that all edges have unique costs; that is, $c(e_1) \neq c(e_2)$ for $e_1 \neq e_2$. Use the analysis of Kruskal's algorithm to argue that there is a unique minimum spanning tree T.

   (c) Let $e \in E$ be a specified edge. Show how to modify Kruskal's algorithm (and its analysis) to compute a minimum spanning tree T subject to the condition that $e \in T$.

   [15 points]

3. Consider the following *interval covering problem*. Input: A set of intervals $\mathcal{I} = \{I_1, \ldots, I_n\}$ where each interval $I_j = [s_j, f_j]$.
   Output: A *cover* $\mathcal{I}' \subseteq \mathcal{I}$ such that for all $I \in \mathcal{I}$, there exists $I' \in \mathcal{I}'$ such that $I' \cap I \neq \emptyset$.

   (a) Show that the "most obvious greedy algorithm" (i.e. always choosing that interval which overlaps the largest number of currently uncovered intervals) does not produce a minimal size cover. [5 points]

   (b) Bonus question: Describe an optimal greedy algorithm for the interval covering problem. Provide a proof that your algorithm always produces an optimal cover. [10 points]