CSC373S **Problem Set 1** **Winter, 2011**

Due: Wed, February 2, 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. (15 points)

   Consider the following circular arc selection problem. We have a circle with circumference c and n arcs $I_1, \ldots, I_n$ on this circle. Every arc $I_j = (s_j, a_j)$ is specified by a starting point s_j and an arc length $a_j < c$ and proceeds in a clockwise direction. So an arc $I_j = (s_j, a_j)$ begins at s_j and ends at $(s_j + a_j) modc$. Describe an $O(n^2)$ algorithm that will select a maximum number of non intersecting arcs.
   Hint: Consider whether or not a particular arc will be in the solution.

   Motivation: You have a PVR that can schedule at most one program at a time and you want to schedule as many daily programs (24 hour clock) as possible.

2. (20 points) Consider the JISP problem introduced in class. Show that the EFT greedy algorithm provides a 2-approximation.
   Hint: You can use a charging argument via a function h that is 2-1; that is, at most 2 intervals in OPT are maped to any interval in the greedy solution.

3. Consider the first fit EFT and best fit EFT greedy algorithms for m-machine interval scheduling.

   (a) (5 points) Show that first fit EFT is not an optimal algorithm.

   (b) (15 points) In the best fit EFT algorithm, show that the partial solution at the end of the i^{th} iteration is promising and thereby conclude that best fit EFT is an optimal algorithm.

4. (15 points)

   Consider the following scheduling problem. The input consists of n jobs with processing times $p_1, \ldots, p_n$. All jobs must be scheduled (on one machine). The goal is to order the jobs so as to minimize the average completion time. If the jobs are scheduled in the order $\pi(1), \pi(2), \ldots, \pi(n)$ then the completion time of job $J(\pi(i))$ is $C_{\pi(i)} = \sum_{k \leq i} p_{\pi(i)}$. Thus the problem is to construct an ordering π so as to minimize $\frac{1}{n} \sum_i C_i = \frac{1}{n} \sum_i C_{\pi(i)}$. Describe an optimal greedy algorithm (i.e. optimal ordering) and prove the optimality of your algorithm using an exchange argument similar to the one used for the minimizing maximum lateness.

5. Consider the following interval covering problem. The input $\mathcal{I}$ is a set of intervals $\{J(1), \ldots, J(n)\}$ with $J(i) = (s_i, f_i)$ (as in interval selection). The goal now is to find a minimum cardinality subset S of these intervals such that every input interval intersects at least one interval in S.

   (a) (10 points) Consider the following greedy algorithm:

   Let $R := \mathcal{I} = \{J(1), \ldots, J(n)\}$ % R will be the intervals not yet covered
   Let $S := \emptyset$ % S will be the desired solution
   While $R \neq \emptyset$
   Let k be the index of an interval in $\mathcal{I}$ that intersects the most other intervals in R % Use any tie breaking rule when there is more than one such k
   $S := S \cup \{J(k)\}$
   remove $J(k)$ and all the intervals with which it intersects from R
   End While

   Show that this algorithm is not an optimal algorithm. What inapproximation ratio can you obtain?

   (b) (Bonus 10 points) Derive a constant approximation bound for this algorithm.

   (c) (Bonus 10 points)

   Provide an optimal greedy algorithm and prove that your algorithm is optimal.

6. (15 points)

   Consider the MST problem for edge weighted undirected graphs. Use your knowledge of Kruskal's MST algorithm to prove the assertion: If all the edge weights are unique, then there is a unique MST.

7. (15 points)

   Consider the following problem relating to paths from a given source node s to all other nodes in edge weighted directed graphs when all edge weights are positive. Let $c(e) = c(u, v)$ denote the cost of edge $e = (u, v)$.

   Define the cost of a path $\pi = v_0, v_1, \ldots, v_k$ to be $min_{i:0 \leq i < k}\{c(v_i, v_{i+1})\}$. The goal now is to find a path from s to each node so as to *maximize* the cost of the path. Show how to use ideas from Dijkstra's algorithm to define an optimal greedy algorithm and sketch a proof that the algorithm is correct (i.e. will compute the optimal path to each vertex).

8. (15 points)

   Modify the DP algorithm for computing the optimal value for the weighted interval selection problem so as to count the number of different optimal solutions $SOL \subseteq \{I_1, \ldots, I_n\}$ for a given instance of the problem. SOL and SOL' are different if they are different as sets.