CS 473: Algorithms, Fall 2009
HW 5 (due Tuesday, October 13th in class)

This homework contains three problems. **Read the instruction for submitting homework on the course web page**. In particular, *make sure* that you write the solutions for the problems on separate sheets of paper. Write your name and netid on each sheet.

Collaboration Policy: For this home work, students can work in groups of up to 3 members each. Each group submits only one written solution (some groups will do an oral presentation. Indicate your group members on the homework (netids are needed)).

1. (30 pts) You are given two sets of intervals one colored red and the other blue. Let $R = \{I_1, \ldots, I_n\}$ be the set of red intervals and let $B = \{J_1, \ldots, J_m\}$ be the set of blue intervals. Each interval I is specified by two integers, $a(I)$ and $b(I)$, the left and right end points of the interval which are part of the interval (that is $I = [a(I), b(I)]$). A red interval I_i *covers* a blue interval J_j if I_i overlaps with J_j. The goal is to choose a smallest set of red intervals from R such that each blue interval is covered by some red interval in the chosen set.

   - Consider the following greedy algorithm. Pick a red interval that covers the largest number of blue intervals, add it to the chosen set, remove the covered blue intervals. Repeat until there are no blue intervals left. Give an example to show that this algorithm does not always produce an optimal solution.
   - Describe an algorithm that finds an optimal solution. Full credit for an algorithm that runs in $O((n+m)\log(n+m))$ time.

2. (30 pts) Consider the following recursive algorithm for computing the edge set of an MST of a given undirected graph $G = (V, E)$ which may have multiple edges between two nodes (a multigraph). Assume that m is the total number of edges and n is the number of nodes. Each node v has all the edges incident to it in a list $Adj(v)$. Assume that the edge costs are distinct and G is connected.

   (a) If $|V| = 1$ output nothing.
   (b) For each $v \in V$, add the cheapest edge incident to v to the set A.
   (c) Let $S_1, \ldots, S_k$ be the connected components induced by the edge set A.
   (d) Obtain a new graph G' from G by shrinking each S_i to a single vertex. There may be multiple edges between two shrunk vertices. Any loops can be removed.
   (e) Recursively find an MST T' in G'.
   (f) Output $T = T' \cup A$ as the MST in G.

   Prove that the above algorithm correctly computes the edge set of the MST of G. Explain how it be implemented to run in $O((m+n)\log n)$ time. *Hint:* What is the size of the instance in the recursive call?

   Extra Credit: (15 points) Modify the above algorithm as follows. Once the size of the graph (in terms of nodes) falls below some threshold, instead of recursing, switch to Prim's

algorithm using Fibonacci heaps. Recall that Prim's algorithm with Fibonacci heaps takes $O(n \log n + m)$ time on a graph with n nodes and m edges. Choose the threshold appropriately to optimize the running time of the resulting algorithm. What is the running time?

3. (40 pts) Explore reductions to known algorithms.

   (a) (20 pts) Let $G = (V, E)$ be an undirected graph with non-negative edge weights. Describe an algorithm that removes the smallest weight subset of edges such that the remaining graph is acylic.

   (b) (20 pts) Consider a variant of interval scheduling except now the intervals are arcs on a circle. The goal is to find the maximum number of arcs that do not overlap. More formally, let C be the circle on the plane centered at the origin with unit radius. Let $A_1, \ldots, A_n$ be a collection of arcs on the circle where an arc A_i is specified by two angles $\alpha_i \in [0, 2\pi]$ and $\beta_i \in [0, 2\pi]$: the arc starts at the point on the circle C with angle α_i and goes counter-clockwise till the point on C at angle β_i (the end points are part of the arc). Two arcs overlap if they share a point on the circle. Describe an algorithm to find the maximum number of non-overlapping arcs in the given set of arcs.