CMPT 307 (Fall 2013)
Quiz 2 (October 29, 2013)
Total marks = 45

A. Questions on intervals (15 points)

1. Let X be a set of intervals on a line. We say that a set P of points *pierce* X if every interval in X contains at least one point of P. Design an algorithm to select a minimum size set of points that pierce X. If you use a greedy algorithm, describe the greedy strategy, and then show that the algorithm produces an optimal solution. What is the running time of the algorithm?

   Ans::

   Step 1: Sort the intervals by their right end points.
   Step 2: Select the leftmost right end point as the first piercing point.
   Step 3: Remove all the intervals that are pierced by this point.
   Step 4: Repeat step 1 through 4 for the remaining intervals.

   Running time is $O(nlogn)$.

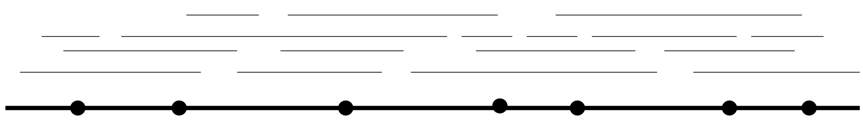
   Proof of optimality: Le $G =< p_1, p_2, \ldots p_k >$ be the piercing set obtained using the above greedy approach where $x(p_i) < x(p_{i+1}) \quad \forall \ 1 \leq i \leq k-1$. Let $O =< q_1, q_2, \ldots, q_t >$ be an optimal piercing set where $x(q_i) < x(q_{i+1} \quad \forall \ 1 \leq i \leq t-1$. Let r be the largest index such that $p_i = q_i, i = 1, 2, \ldots r$, and $p_{r+1} \neq q_{r+1}$. Let X' be the subset of X pierced by $p_1, p_2, \ldots, p_r$. Since there is no right endpoint of the intervals of X' in the open interval (p_r, p_{r+1}) (greedy approach guarntees that), we can move q_{r+1} to position p_{r+1} and still pierce the same set of intervals pierced by q_{r+1}. Thus $O' =< q_1, q_2, \ldots, q_r, p_{r+1}, q_{r+2}, \ldots, q_t¿$ is also an optimal solution. Continuing like this we will be able to show that G is an optimal piercing set of X.

2. Consider the problem of scheduling lectures in classrooms. Each lecture I is represented by an interval $[a, b]$ where a is the starting time and b is the ending time of lecture I. Two overlapping intervals cannot be schuduled in the same classroom at the same time. Answer the following questions based on the figure below.

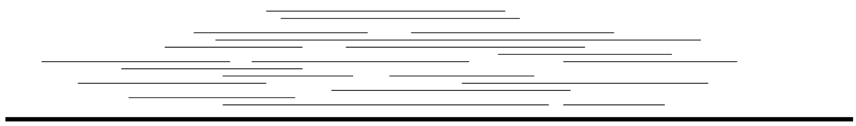
   (a) What is the lower bound on number of classroms needed to schedule the lectures in the following figure?
   (b) What is the upper bound?
   (c) Explain in a few sentences why your answer is the best possible.

Ans:

10 is the lower bound. This is because at any instance of interval partitioning, the number of resources required is at least the depth of the set of intervals. We can come up with a greedy solution (page 124, Kleinberg and Tardos(KT)) that assigns a label to intervals from a set of d labels , d equalling the depth, such that overlapping intervals are labeled differently. We can always find a solution that uses number of resources equal to depth. So 10 is also the upper bound.



A−1. Seven points pierce all the intervals.



A−2. Set of lectures. Each interval represents the length of the lecture.

Figure 1:

B. Questions on Minimum Spanning Tree (15 points)

1. Describe the cut and the cycle properties in a graph $G = (V, E)$, related to MSTs.

   Ans:: Cut Property: See definition 4.17, KT page 145. Cycle Property: See definition 4.20, KT page 147.

2. Consider the following algorithm, called reverse-delete algorithm, to compute MST of an undirected graph $G = (V, E)$ where the edges are weighted.

   Edges are considered in descending order of weight. For each edge being considered, remove the edge from G unless it would disconnect the graph.

   (a) Show that the reverse-delete algorithm computes an MST.

(b) How quickly can you implement the algorithm? Assume the the graph is represented by an adjacent list.

Ans:

a. See the proof 4.21, (KT page 148).

b. The edges have to be first sorted in descending order.Whenever an edge is considered, we should check to see if removing it would disconnect the graph. This can be done with BFS, $O(|m| + |n|)$. Since this step has to be performed for all the m edges, our implementation time for reverse delete is $O(m^2)$.

[Though it should be noted that with a faster connectivity check, reverse delete can be done in $O(mlogn(loglogn)^3)$]

C. Questions on shortest paths (15 points)

1. Consider a directed graph in which the only negative edges are those that leave s; all other edges are positive. Can Dijkstra's algorithm, started at s, fail on such a graph? Explain.

   Ans: Yes it can fail. We can easily come up with an example involving a negative edge leaving s and a positive edge returning to s, and part of a negative cycle. An example was discussed in the class.

2. You are given a set of cities, along with the pattern of highways between them in the form of graph $G = (V, E)$. Each stretch of highway $e \in E$ connects two of the cities of l_e km distance apart. You want to go from city s to city t. There is one problem: your car can only hold enough gas to cover L kilometers. There are no gas stations between cities. Therefore, you can only take a route if every one of its edges has length $l_e \leq L$. Given the limitation on your car's fuel tank capacity, show how to determine in $O(|V| + |E|)$ time whether there is a feasible route from s to t. Provide the implementation details.

   Ans: Since we can only take a route if every one of its edges has a length less than or equal to L, we can simply remove edges of a greater value and see if the city t is still reachable from s. A simple Breadth First Search should be able to verify this. BFS step is $O(|V| + |E|)$.