CSE548/AMS542 Fall 2007 Analysis of Algorithms

Midterm

- This is a closed book exam. You can bring the TCS cheat sheet however.
- There are 5 problems and 30 points total.
- You can refer to the algorithms we have covered in class without referring to the details. If you give a greedy algorithm but do not give the proof of its correctness, you may lose all (if your algorithm is wrong) or partial points (if your algorithm is correct).
- In class on Friday, October 26th, 2007. Exam starts at 9:35am and ends at 10:55am sharp.

By signing below I declare that I follow the rule of academic integrity and finish the exam on my own, without the help of others.

Name ____________________________

ID ____________________________

Signature ____________________________

Score	Grade
1	
2	
3	
4	
5	
Total	

1. **Short questions** (5pts) Answer the following questions. Simply give the **final** answer. You **do not** need to give proofs or counter-examples.

   (a) In a graph with distinctive weights (no two edges have the same weight), the edge of the minimum weight on a cycle is on the minimum spanning tree. True or false?

   (b) In a graph with distinctive weights (no two edges have the same weight), the shortest path between any two nodes is unique. True or False?

   (c) If an undirected graph has some edges with negative weights, then are the following 2 claims true or false? Claim 1: Dijkstra's algorithm definitely will output the single-source shortest path tree, in any cases. Claim 2: Dijkstra's algorithm definitely will **not** output the single-source shortest path tree, in any cases.

   (d) What is the solution to the recurrence $T(n) = 2T(n/4) + \sqrt{n}$?

   (e) What is the solution to the recurrence $T(n) = T(n-1) + 1/n + \sqrt{\log n}/n^2$?

2. **Second-best minimum spanning tree** (10pts). Given an undirected graph $G = (V, E)$ with distinct weights on its edges (no two edges have the same weight). The minimum spanning tree is the tree with minimum total weight. The second-best minimum spanning tree has exactly one tree with smaller total weight.

   (a) Suppose T is the minimum spanning tree and T' is the second-best minimum spanning tree. Show that T' and T differ by exactly one edge. That is, T and T' agree on $n - 1$ edges. (5pts)

(b) Find an efficient algorithm to compute the second best minimum spanning tree and give its worst-case running time. Try to find the most efficient algorithm you can. If you get an algorithm with $O(m \log n)$ running time then you get full points. (5pts)

3. **Distances between polygonal curves** (4pts). A polygonal curve of n vertices $x_1, x_2, \cdots, x_n$ is the concatenation of $n-1$ line segments $x_i x_{i+1}$, for $i = 1, 2, \cdots, n-1$. Given two polygonal curves X_1, Y_2, each with n vertices $\{x_i\}$, $\{y_i\}$ respectively, a natural question is to define the distance between them.

   The Frechét distance is defined in the following way. Imagine that one person walks on the curve X and a dog walks on the curve Y. The dog is tied with a leash (or rope). Both the person and the dog can not walk backward. In addition, to simplify the problem, the person and the dog can only walk one at a time. When the person (or the dog) walks, the dog (or the person) needs to wait at a vertex. The Frechét distance is the minimum length of the leash for the person and the dog to possibly finish the walk.

   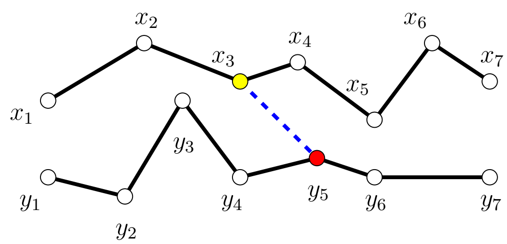

   Figure 1: Frechet distance.

   Give an $O(n^2)$ algorithm to compute the discrete Frechét distance of the two curves. (5pts)

4. **Manhattan skyline** (5pts) Given a set of n rectangles, each rectangle i represented by its left and right x-coordinates a_i and b_i and its height h_i. The bottom of each rectangle is on the x-axis. Design an $O(n \log n)$ algorithm to compute the vertices of this skyline.

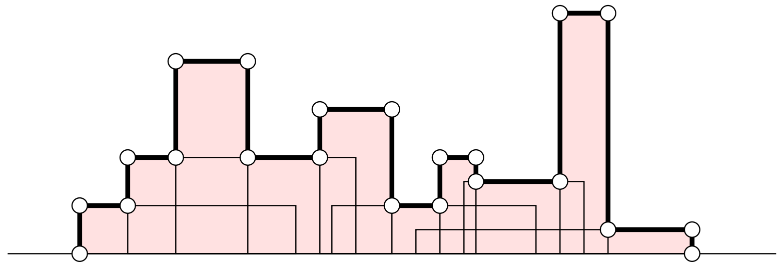

Figure 2: Manhattan skyline.

5. **Job scheduling** (6pts) You are given a set of n jobs. Job i starts at time s_i and ends at time t_i with a profit of f_i. You can only take one job at a time. The goal is to choose a subset of jobs to maximize the profit.

   (a) Show that the following algorithm does not give the optimal solution: sort the jobs with their profits in decreasing order, $f_1 \geq f_2 \geq f_3 \geq \cdots \geq f_n$. Try to include each job in the schedule in turn. If job i does not conflict with any jobs you have taken so far, include it in the schedule, otherwise discard it. (2pts)

(b) If the jobs have profit of either 1 or 2 (i.e., $f_i \in \{1, 2\}$), give a greedy algorithm to find a subset of jobs with total profit at least 1/2 of the profit of the optimal solution. (Hint: use the greedy interval scheduling algorithm.) (4pts)