CS 6550 – Design and Analysis of Algorithms – Fall 2007
Final – Due December 6.

You must work alone on this final. Please feel free to use the lecture notes on the web or your text book, and of course you can contact Shiva or me for clarification.

1. Let P be a set of n points in the plane and let $p \in P$ be one of these points. Give a randomized algorithm that can decide in $O(n)$ expected time whether p is a vertex of the convex hull of P.

2. We have a graph $G = (V, E)$ with n vertices. Now suppose we would like to find a collection of matchings such that every edge of the graph is a member of (at least) one of the matchings selected. The goal is to pick the minimum number of matchings for our collection. Give an $O(\log n)$-approximation algorithm for this problem.

3. With elections around the corner, suppose that you want to estimate the fraction f of Democrats in Georgia. Assume that you are able to select a resident uniformly at random and determine their political affiliation. Assume also that you know some lower bound $a < f$. Devise a procedure for estimating f by some $\widehat{f}$ such that $Pr[\{|f - \widehat{f}| > \epsilon f] < \delta$, for any choice of constants $0 < a, \epsilon, \delta < 1$. Determine the number of residents you must query to get the estimate.

4. Let $S = \{p_1, \ldots, p_n\}$ be an input set of n points in the plane. Suppose our goal is to find a pair of points p_i and p_j that are as far apart as possible, maximizing the Euclidean distance between points p_i and p_j among pairs $\{i, j\}$. Give an efficient method for solving this problem. You should be able to do it in worst-case time $O(n \log n)$.

5. You are given a universe U of n elements. For each $i = 1, \ldots, m$, you are given two subsets $S_i, T_i \subseteq U$. Your goal is to select exactly one of S_i or T_i, for each i, while minimizing the size of the union of the sets selected. As an application, imagine that you need to do m jobs, and job i can be performed either by team S_i or by team T_i. Your goal is to hire as few people as possible, while ensuring that all jobs can be performed with subsets of the teams you hired.

   Give (and analyze) a polynomial-time 2-approximation algorithm for this problem. (Hint: formulate an LP and round it appropriately.)

6. You are watching a stream of packets go by one at a time, and want to take a random sample of k distinct packets from the stream. You face several problems:

   - You only have room to save k packets at any one time.
   - You do not know the total number of packets in the stream.
   - This is an online problem, so If you choose not to save a packet as it goes by, it is gone forever.

Devise a scheme so that, whenever the packet stream terminates, you are left holding a subset of k packets chosen uniformly at random from the entire packet stream. If the total number of packets in the stream is no more than k, then you should hold all of those packets.

7. Consider the following game. You are given an n-sided die that comes up with a number in the range $k \in \{1, ..., n\}$ uniformly at random. Subsequently, you are given a k-sided die. This continues until you have a 1-sided die, and the game ends.

   a) What is the expected number of times that you will have to roll a die before the game ends?

   b) What can you say about the number of times one has to roll the dice so that the game ends with probability $1 - 1/n$?

8. We are given a bipartite graph with men on one side and women on the other. Only the women are known to us in advance, while the men arrive one by one. As a man arrives, his incident edges are revealed and he can be matched off to any of the woman with whom he shares an edge.

   (a) Give a deterministic online bipartite matching algorithm that has a competitive ratio of 2. Show your analysis clearly.

   (b) Give a worst case example, showing your analysis is tight.

   (c) Now consider the following algorithm. As a man arrives, select a random node adjacent to him (that is still available) and match him to her. What is the competitive ratio of this algorithm? Give a tight example.