Final Exam, CS280A

Turn in: noon, Dec 11, 2000

Do all four problems below. Turn in your solutions (by noon Monday, Dec 11) either to my office, BH 3731F, or to my mailbox in suit BH 3731. Do not submit solutions by email. Absolutely no late solutions will be accepted.

You don't have to type your solutions, but being CS graduates with easy access to computers and printers, it will be puzzling why wouldn't you. In any case if you do write your solutions, pls write clearly.

You are *not* allowed to use any source other than your textbook and your notebooks from the lectures. Also, you are not allowed to cooperate.

Be precise, formal, rigorous. Prove everything. **Explain every algorithm with words**, define the symbols you use precisely, and put comments in your "code". You don't have to code your algorithms, but your algorithm must be obvious from your presentation. It doesn't pay to be vague when you don't have an exact algorithm in mind!.

For every algorithm, *prove carefully its correctness and analyze its running time.*

Partial answers will receive partial credit. Less efficient algorithms also get credit.

If you need to make any extra assumptions (e.g. that all points are distinct), state your assumptions clearly.

Problem 1. Given a collection of sets $S_1, S_2, \ldots, S_k$, a subcollection $U = \{S_{i_1}, \ldots, S_{i_B}\}$ is called *witness* if $\cup U$ has at least one common point with each S_i. In the WITNESS problem the input consists of a collection of sets and a bound B and the question is whether there is a witness set of size B or less.

1. Show that the WITNESS problem is NP-complete.
2. Give an efficient approximation algorithm when the sets S_i have size at most 3.

Problem 2. Consider the following transformation of a string x: Choose three consecutive symbols of x and change them. For example a**baa**ba can be transformed to a**aab**ba by changing the second, third, and fourth symbol.

Given two strings x and y of length n we define their distance as the minimum number of transformation to transform x to y. Give an efficient algorithm that takes as input two strings x and y and computes their distance. Prove carefully the correctness of your algorithm.

Problem 3. Recall that a polygon is given by ordering its vertices.

1. Give an algorithm to answer whether a point is inside or outside of convex polygon of n vertices in time $O(logn)$.
2. Give an $O(n + m)$ algorithm to find the intersection of two convex polygons, one of n vertices and the other of m vertices.
3. Give an $O((n+k)logn)$ algorithm to find all the k intersections between n line segments.
4. Given n points $x_1, x_2, \ldots, x_n$ in the plane, let $S_i = \{p | \forall j,\ d(p, x_i) \leq\ d(p, x_j)\}$, where $d(w, z)$ is the Eucledean distance between w and z. Prove that S_i is a covex polygon.
5. The collection of all $S_i,\ i = 1, \ldots, n$ is called the Voronoi diagram of $x_1, \ldots, x_n$. Prove that the Voronoi diagram has at most $O(n)$ vertices and edges.
6. Prepare a data structure of atmost $O(n^2)$ space, such that a query about point p returns the set S_i at which p resides (if p is on the boundary of such sets return an arbitrary one), takes $O(logn)$ times. (Hint: Line Sweep)

Problem 4.

1. Consider an n-element list in n-processor EREW parallel random-access machine, where some elements of the list are marked as blue. Describe an efficient parallel algorithm to form a new list consisting of all the blue elements.

2. Suppose that some nodes in an n-node binary tree are marked blue. Describe an efficient EREW algorithm to form a list consisting of the blue nodes that do not have a blue ancestor.