Spring 1996 **CMSC 754: Homework 4** Samir Khuller

Due in class: Apr 3.

(1) The algorithm described below claims to find the convex hull of a simple polygon. Find the flaw in the algorithm by making a counter-example. Is there a simple description of the kind of polygons for which this algorithm might work? The algorithm is essentially doing a Graham scan on the vertices of the simple polygon, in the order in which they appear. Assume that no three points are collinear.

Let Q be a "stack" initialized as (q_0, q_1) where q_0 is the leftmost point and q_1 is the clockwise successor. March around the polygon starting with the clockwise successor of q_1. If at any stage the stack is $(q_1, q_2, \ldots, q_i)$ and p is the sucssessor of q_i. If $q_{i-1}q_ip$ is a "right" turn then push p on Q; otherwise pop Q repeatedly, until this right turn condition is true, and then push p on Q. Stop when you return to q_0.

(2) Problem 3 in Exercises 3.7.1 (pg 101).

(3) A ski instructor has n pairs of skis to assign to n skiers. The skis have lengths $\ell_i, (1 \leq i \leq n)$ and the skiers have heights $h_j, (1 \leq j \leq n)$. How should the instructor assign skis to skiers so that the resulting sum of absolute differences between height of skier and assigned ski length is as small as possible? Your task is to *design* an $O(n \log n)$ algorithm and *prove* its correctness.
(Note that *exactly* one pair of skis is assigned to each skier.)

(Extra Credit) This can be viewed as a "matching" problem for points on a straight line. How does one generalize the solution for the case when the number of skiers is smaller than the number of skis?

(Extra Extra Credit) Can we get an $O(n^2)$ algorithm for matching points on the plane? In other words, given n points in the plane, how does one find a minimum weight perfect matching quickly? Weights are assumed to be Euclidean distances between points. (I think this problem is still open.)

(4) (Dynamic maintenance of Voronoi Diagram's)

- Given a voronoi diagram of a set S of n points in the plane, show how to delete a point $y \in S$, thereby constructing the voronoi diagram of $S \setminus y$. Your method should take $O(k \log k)$ time where k is the number of edges on the boundary of the voronoi region corresponding to y.
- Given a voronoi diagram of a set S of n points in the plane, show how to add a point $y \notin S$, thereby constructing the voronoi diagram of $S \cup \{y\}$. Your method should take $O(\log n + k)$ time, where k is the size of the "change". (You may assume that along with the new point you are also given the voronoi region(s) it belongs to.)