COS 423: Homework Assignment

due Wed 03/14/2012

Problem 1

We showed in class how to use the conjugation tree as a BSP tree in two dimensions so that halfplane range queries on n points can be answered in time $O(n^{0.695})$. Suppose the query is a triangle and you want to count how many points fall inside it. Can you do it in $O(n^{0.695})$? What if the query is a convex polygon with k sides? Since k can be as large as n, the answer is obviously no. Then how fast can you do it as a function of n and k. In all cases, the search algorithm is the same: visit every node whose region intersects the query polygon but do not pursue the search below any node whose region is enclosed in the polygon.

Problem 2

In the analysis of the conjugation tree, we were careful to base the recursion on one child and one grandchild: $Q(n) = Q(n/2) + Q(n/4) + O(1)$. We could also have initiated the recursion with the three grandchildren, as in: $Q(n) = 3Q(n/4) + O(1)$. What would be the bound then?

Problem 3

Consider a balanced binary tree with n nodes. Pick k leaves and let T be the set consisting of all of their ancestors. The size of T is at most $k \log n$. Give an example for which this bound is far for being tight (for arbitrarily large values of n). Give an asymptotic bound that is tight as a function of k and n. (Big-oh notation is fine.)

Problem 4

Given a set of n points in general position (ie, no three points are collinear; no two of them share the same coordinates), our goal is to bound the set of bisectors. This is an extraordinarily difficult problem, which is still open, so we will seek only a crude, conservative upper bound. A *bisector* is a line passing through two of the n points with exactly $n/2 - 1$ points on each side; assume that n is even. Sort the points by x-coordinates and form groups $S_1, \ldots, S_k$ of the same size (except perhaps for the last one S_k): the points of S_i lie within the i-th vertical strip from the left.

1. Find a (naive) upper bound on the number of bisectors with both points in the same group S_i.

2. Fix an index i and prove that the number of bisectors with their two points on both sides of a vertical line L_i separating the groups $S_1, \ldots, S_i$ from $S_{i+1}, \ldots, S_n$ is $O(n)$. Hint: Rotate a line L slowly clockwise while maintaining the bisecting property. The line will "roll" around

various convex hulls on the left and right sides of L_i while losing and gaining points as it moves. Try to understand how the number of points lying in the four wedges defined by L and L_i evolves.

3. Use your bounds in (1,2) to form an upper found $f(n,k)$ on the total number of bisectors. Set k to maximize that function. What do you get? (Big-oh notation is fine.)

Problem 5

In the low-cutting polygon construction, we tried to match as many points as we could without getting hit by that nasty square-root denominator. What if we stop the matching earlier and also change the weight function? Why double and not quadruple or multiply by some parameter of your choice? Play around with these ideas and see if you can come up with interesting statements about low-cutting constructions involving subsets of the points: perhaps something along the lines of, Given any set of n points in general position, there always exists a subset of size X yielding a matching or a spanning tree or a spanning polygon of cutting number Y...