Massachusetts Institute of Technology
6.854J/18.415J: Advanced Algorithms
Erik Demaine and David Karger

Handout 22
Wednesday, November 12, 2003

Problem Set 11

Due: Monday, November 17, 2003.

Problem 1. Describe a randomized incremental algorithm to compute the intersection of half-planes and analyze its expected running time. A half-plane is given by an inequality of the form $ax + by \leq c$.

Your algorithm should maintain the intersection of the current set of half-planes. To figure out where to insert a new half-plane, maintain a *conflict graph* between the vertices of the current intersection and the half-planes that are still to be inserted. A conflict graph is a bipartite graph where L is the set of the half-planes to be inserted, and R is the set of vertices of the current intersection. The nodes of the bipartite graph are $L \cup R$. There is an edge between $l \in L$ and $r \in R$ if r is in the complement of the half-space l.

Problem 2. The randomized incremental construction algorithm for convex hull presented in class was shown to run in $O(n \log n)$ *expected* time. In this problem, we show that it actually runs in $O(n \log n)$ time with high probability (probability $1 - O(1/n^c)$ for any desired $c > 0$).

Define an indicator random variable $X_i(p)$ to be 1 if p's pointer is changed in the ith step. Recall that in the algorithm, each point p maintains a pointer to the edge on the convex hull that intersects the ray from p_0 (an internal point in the convex hull) to p. Show that, for each point p, $\sum X_i(p)$ is $O(\log n)$ with high probability so that the total work is $O(n \log n)$ with high probability.

Problem 3. Recall the range-tree data structure seen in class. Given a set P of n points in the plane, we can build a range tree to answer queries of the form: which points in P lie in a given rectangle? The query time is $O(\lg^2 n + k)$, where k is the number of points in the rectangle, construction time is $O(n \log n)$, and the space is $O(n \log n)$. The range tree consists of a primary structure of a balanced binary search tree on the x-coordinate of the points in P. Each internal node v of the primary tree has an associated *secondary structure* of a balanced binary search tree on the y-coordinates of the leaf nodes in the subtree rooted at v.

In this problem, we consider one possible approach to lowering the space requirements of range trees.

(a) Suppose that only the nodes with even depth have an associated *secondary structure*. Show how the query algorithm can be adapted to answer queries correctly.

(b) Analyze the storage requirements and query time of this modified data structure.

(c) Suppose that only the nodes with depth $0, \lfloor \frac{1}{j} \log n \rfloor, \lfloor \frac{2}{j} \log n \rfloor, \ldots$ have an associated *secondary structure*, where $j \geq 2$ is a constant. Analyze the storage requirements and query time of this data structure. Express the bounds in terms of n and j.