Advanced Algorithms, Fall 2011

Homework #7, due Monday, November 28

1. Given an undirected graph $G = (V, E)$, with $|V| = n, |E| = m$, consider the following method of generating an independent set. Given a permutation σ of the vertices, define a subset $S(\sigma)$ of the vertices as follows: for each vertex $i, i \in S(\sigma)$ if and only if no neighbor j of i precedes i in the permutation σ.

   (a) Show that each $S(\sigma)$ is an independent set in G.

   (b) Design a randomized algorithm to produce σ with $|S(\sigma)| = \Sigma_{i=1}^{n} \frac{1}{d_i+1}$ where d_i is the degree of vertex i.

   (c) Prove that G has an independent set of size at least $\Sigma_{i=1}^{n} \frac{1}{d_i+1}$.

2. Consider the following experiment that proceeds in a sequence of rounds. For the first round, we have n balls, which are thrown independently and uniformly at random into n bins. After round i, for $i \geq 1$, we discard every ball that ended up in a bin by itself in round i. The remaining balls are retained for round $i+1$, in which they are again thrown independently and uniformly at random into the n bins.

   (a) If in some round there are εn balls, how many balls would you expect to have in the next round?

   (b) Assuming that everything proceeded according to expectation, prove that we would discard all the balls within $O(\log \log n)$ rounds.

   (c) Prove that with probability $1 - o(1)$, the number of rounds is $O(\log \log n)$.

   Hint: call a round *good* if the number of balls retained is not much more than expected. Compute the probability that a round is good. Show that with probability $1 - o(1)$, we get enough good rounds among the first $O(\log \log n)$ to finish.

3. You are given a collection S of segments in the plane. Assume that no segment is degenerate or horizontal, that no two segments are collinear, and that no two segments intersect. Build a binary search tree using these segments as follows:

```
TreeBuild(S)
if |S|=0
   then tree T is a single node
   else select s uniformly at random from S
        /* denote by l(s) the line subtending s */
        /* l(s) splits the current area and
           may intersect remaining segments */
        /* denote by l(s)- and l(s)+ the left and right half-planes,
           not including the line l(s) itself */
        let S+ contain the intersections of elements of S with l(s)+
        /* note: each intersection is an intersection of geometric
           objects -- a half-plane and a segment -- so the result
           is a geometric object: a segment, a piece of a segment,
           or nothing */
        T+ <- TreeBuild(S+)
        let S- contain the intersections of elements of S with l(s)-
        T- <- TreeBuild(S-)
        let T have s at the root, T- as left subtree, and T+ as right subtree
return T
```

The tree leaves correspond to convex regions, each of which is empty of any segment—all segments and segment pieces are on boundaries.

Analyze this algorithm in terms of both expected and worst-case running times. (Hint: first derive the expected and worst-case number of pieces of segments created by the splitting process, then use these values to bound the expected and worst-case size of the tree, then finally analyze the running time.)

4. (a) Let S be a set of elements, $\oplus$ an associative and commutative binary operation on S, and assume that the product of an element of S by a positive real value is well defined. Prove that, if S_1 and S_2 are convex subsets of S—that is, if, $\forall x, y \in S_i$ and $\forall \alpha$, $0 \leq \alpha \leq 1$, we have $\alpha \cdot x \oplus (1-\alpha) \cdot y \in S_i$ (for each $i = 1, 2$)—, then $S_1 \cap S_2$ is also convex.

   (b) Now consider the intersection of a set S of halfplanes in 2D. From the first part of this question, this intersection is a convex polygon (possibly unbounded). Show how to compute the intersection in $O(|S| \log |S|)$ time.

   (c) If that intersection polygon has n edges, then only n halfplanes are needed to define it; any halfplane of S that does not contribute an edge to intersection polygon is called *redundant*. Prove that every redundant halfplane contains the intersection of two halfplanes from S (possibly themselves redundant).

   (d) If the intersection polygon has n edges, is it possible to compute it in $O(n \log |S|)$ time? in $O(|S| \log n)$ time? If the worst-case remains unchanged, would randomization help?