Advanced Algorithms, Fall 2011

Homework #9, due Monday, December 12

1. Devise and analyze an algorithm that uses randomized incremental construction to compute the intersection of a collection of halfplanes in 2D.

2. In 2D computational geometry, a *chord* of a simple polygon is an edge between nonadjacent vertices of the polygon. Two distinct chords, (w,x) and (y,z) are nonintersecting if there is a path of polygon edges from w to x that does not contain either y or z as an intermediate vertex. If two chords share exactly one vertex, they are deemed nonintersecting. Each chord has a weight associated with it.

   Design an algorithm (DP) that, given a polygon and its set of chords and associated weights, finds a maximal subset of mutually nonintersecting chords such that the sum of chord weights in the subset is minimized.

3. You have to place several advertisements for your company in a TV channel on a specific day. The possible time slots for the advertisements are t_1, t_2, ..., t_n; each slot t_i has its own expected revenue r_i. You have to choose the time slots in order to maximize the total expected revenue (which is the sum of revenues of the time slots chosen) with the restriction that no two advertisements must be occur within T time units of each other.

   Design a dynamic programming solution for this problem. Does a greedy strategy return an optimal solution? (Define your choice of greedy; then either prove that it is optimal or give a counterexample.)

4. A biotechnology company has sequenced a long sequence T (of length m). They also have a large library of shorter sequences, each of length at most n, $n \ll m$.

   Design an algorithm (DP) to select library sequences $l_1, l_2, \ldots, l_k$ such that their concatenation $S = l_1 l_2 \ldots l_k$ has the minimum alignment cost (for any given cost model) with the given long sequence T. Note that repetitions are allowed—that is, $i \neq j$ need not imply $l_i \neq l_j$.