CS 473: Undergraduate Algorithms, Spring 2009
HBS 2

1. Consider two horizontal lines l_1 and l_2 in the plane. There are n points on l_1 with x-coordinates $A = a_1, a_2, \ldots, a_n$ and there are n points on l_2 with x-coordinates $B = b_1, b_2, \ldots, b_n$. Design an algorithm to compute, given A and B, a largest set S of non-intersecting line segments subject to the following restrictions:

   (a) Any segment in S connects a_i to b_i for some $i(1 \leq i \leq n)$.

   (b) Any two segments in S do not intersect.

2. Consider a $2^n x 2^n$ chess board with one (arbitrarily chosen) square removed. Prove that any such chessboard can be tiled without gaps or overlaps by L-shaped pieces of 3 squares each. Can you give an algorithm to do the tiling?

3. Given a string of letters $Y = y_1 y_2 \ldots y_n$, a segmentation of Y is a partition of its letters into contiguous blocks of letters (also called words). Each word has a quality that can be computed by a given oracle (e.g. you can call *quality("meet")* to get the quality of the word "meet"). The quality of a segmentation is equal to the sum over the qualities of its words. Each call to the oracle takes linear time in terms of the argument; that is quality(S) takes $O(|S|)$.

   Using the given oracle, give an algorithm that takes a string Y and computes a segmentation of maximum total quality.