Computational Geometry (CS 235)	Prof. Suri
Homework Assignment 1	
Handed Out: Oct. 7	*Due: Oct. 14*

In all problems, carefully but concisely explain your algorithm, and always justify its correctness and complexity. Make your answers short and concise, and preferably typeset your answers for ease of reading.

1. (10 pts) Let P be a convex polygon with n vertices $(p_1, p_2, \ldots, p_n)$ given in cyclic order, in an array. Describe a scheme to preprocess the polygon using $O(n)$ space and $O(n)$ time, so that the following query can be answered in $O(\log n)$ time:

   given a line ℓ, does ℓ intersect P and, if so, determine the intersection points $\ell \cap P$?

2. (30 pts) [**Chan Revisited:**]
   Recall that in Chan's optimal convex hull algorithm, the algorithm "hunts" for the convex hull size h, by using a parameter m, which increases as follows: $m = \min(2^{2^t}, n)$, for $t = 1, 2, \ldots$.

   Suppose, instead, Chan had chosen to do the following:

   - $m = \min(2^t, n)$, for $t = 1, 2, \ldots$.
   - $m = \min(2^{2^{2^t}}, n)$, for $t = 1, 2, \ldots$.

   What is the run time complexity of Chan's algorithm for these choices of the search function.

3. (30 pts) You are given a set S of n vertical line segments in the plane. A **stabber** for S is a line that intersects all segments in S. Describe an $O(n \log n)$ time algorithm to decide if there is a stabber for the given set S.

4. (30 pts) Consider the following variant of the divide-and-conquer convex hull algorithm, which computes the upper hull $U(P)$ of an n-point planar point set P.

(a) If $|P| \leq 3$, compute the upper convex hull by brute force in $O(1)$ time and return.

(b) Otherwise, partition P into two sets A and B, where A is the left half and B is the right half of the points (in the order determined by their x-coordinates).

(c) Compute the *upper tangent* T between A and B, and let $a \in A$ and $b \in B$ be the end points of this tangent. Note that T is being computed **without** knowing the convex hulls of A and B!

(d) Remove from A and B all the points whose x-coordinates lie strictly between a_x and b_x. (We do not remove a or b.) Let $A' \subseteq A$ and $B' \subseteq B$ be the remaining point sets.

(e) Recursively compute the upper hulls $H(A')$ and $H(B')$.

(f) Return as upper hull $H(A)$ the concatenation of $H(A')$ followed by tangent ab, followed by the upper hull $H(B')$.

The main and very important difference between this algorithm and the divide-and-conquer presented in class is that this algorithm computes the tangent between the two halves **before** it recursively computes the hulls of the left and the right halves.

You may assume (without proving) that the tangent computation of Step (c) can be done in linear time (linear in the size of the problem).

With this assumption, show that this algorithm runs in worst-case time $O(n \log h)$, where h is the number of points in the final upper convex hull of P.