CSc 437
Answers to homework #2

Instructions. All assignments are to be completed on separate paper. Use only one side of the paper. Assignments will be due at the beginning of class, or via email. To receive full credit, you must show all of your work. You can email the homework to the grader.

All questions are taken from the textbook

1. 3.1
2. 3.3
3. 3.6
4. 3.10 (start by computing the convex hull, and split the resulting polygon into sub-polygons)

   Answer: *There are many ways to solve this question. We need to compute in time $O(n \log n)$ a triangulation of a set S of n points. Here is one: We first (in time $O(n \log n)$) compute $CH(S)$. Let C denote the set the vertices of $CH(S)$. We sort the points of $S \setminus C$ by their y-value. Let $p_1 \ldots p_k$ be the resulting set, in a increasing order of their y-value, (so p_1 is the lowest one). We connect p_i to p_{i+1}, forming a y-monotone path. We connect the lowest point of C to p_1, and p_k to the heights point of C. This split $CH(S)$ into two y-monotone polygons. We use the algorithm studied in class to triangulate each other them.*

5. Assume $h_1 \ldots h_n$ are halfplanes in 2D, given in increasing slopes of their bounding lines. Compute $h_1 \cap h_2 \cap \ldots h_n$ in $O(n)$ time.

   Answer:

   We split the set into two subsets, ones that containing the point $(0, \infty)$, and ones that contains $(0, -\infty)$. So from now on we assume that all constrains contain the point $(0, -\infty)$. Let

   $$P(i) = \bigcap_{j=1}^{i} h_i$$

   . We maintain $P(j)$ by maintaining its boundary, which is a polygonal chain, which is a part of a convex polygon. Let $L(i)$ denote this chain. Note that the first and last edges of $L(i)$ are unbounded. Note (check) that once h_{i+1} in added to form $P(i+1)$, it crosses $L(i)$ in exactly one point. To find this point, we scan each edge from the rightmost one of $L(i)$, until finding the intersection point. Note that each edge that is scanned is contributed by a constrains that would not appear in future $P(i')$ (for $i' > i$). Hence the number of edges that is scanned over the

whole course of the algorithm cannot exceed the number of constrains, which is n. Checking with a edge of $L(i)$ is crossed by the line bounding h_i takes $O(1)$. Thus the running time is $O(n)$.

6. It is known that the union of n axis-parallel squares in the plane has complexity $O(n)$. Show how to compute their complexity in time $O(n \log^2 n)$ (use divide and conquer).

   Answer: *We combine the paradigm of line-sweep with the paradigm of divided and concur. Let S be the set of n unit axis-parallel squares. We split S into S_1 and S_2, two subsets that contains $n/2$ squares. We compute recursively U_1 and U_2, which is the union of S_1 and of S_2. Recall that the complexity of S_1 (and of S_2) is $\leq Kn/2$, for a constant K. Next we use a line sweep to compute their union. Note that every event that the sweeping line encounter is either a vertex of U_1, or a vertex of U_2, or an intersection between an edge of U_1 and an edge of U_2, but in the later case, this vertex (prove) must be a vertex of $U_1 \cup U_2$. Also recall that the number of vertices of $U_1 \cup U_2$ is $\leq Kn$. Hence the total number of events is at most $Kn/2 + Kn/2 + Kn$. The running time of the merging process is therefore $O(n \log n)$, and the running time of the algorithm is*

   $$T(n) = 2T(n/2) + 3Kn \log n$$

   whose solution is $O(n \log^2 n)$.