Please work on these problems and be prepared to share your solutions with classmates in class. Assignments will not be collected for grading.

1. The convex hull of a set $S$ is defined to be the intersection of all convex sets that contain $S$. For the convex hull of a set of points it was indicated that the convex hull is the convex set with smallest perimeter. We want to show that these are equivalent definitions.

   a. Prove that the intersection of two convex sets is again convex. This implies that the intersection of a finite family of convex sets is convex as well.

   b. Prove that the smallest perimeter polygon $P$ containing a set of points $P$ is convex.

   c. Prove that any convex set containing the set of points $P$ contains the smallest perimeter polygon $P$.

2. The $O(n \log n)$ algorithm to compute the convex hull of a set of $n$ points in the plane that was described in this chapter is based on the paradigm of incremental construction: add the points one by one, and update the convex hull after each addition. In this exercise we shall develop an algorithm based on another paradigm, namely divide-and-conquer.

   a. Let $P_1$ and $P_2$ be two disjoint convex polygons with $n$ vertices in total. Give an $O(n)$ time algorithm that computes the convex hull of $P_1 \cup P_2$.

   b. Use the algorithm from part a to develop an $O(n \log n)$ time divide-and-conquer algorithm to compute the convex hull of a set of $n$ points in the plane.

The next question asks about the string wrapping algorithm that we discussed in class. We probably don’t need to discuss this further, nevertheless I though it would be interesting to include this.

3. Consider the following alternative approach to computing the convex hull of a set of points in the plane: We start with the rightmost point. This is the first point $p_1$ of the convex hull. Now imagine that we start with a vertical line and rotate it clockwise until it hits another point $p_2$. This is the second point on the convex hull. We continue rotating the line but this time around $p_2$ until we hit a point $p_3$. In this way we continue until we reach $p_1$ again.

   a. Give pseudocode for this algorithm.

   b. What degenerate cases can occur and how can we deal with them?

   c. Prove that the algorithm correctly computes the convex hull.

   d. Prove that the algorithm can be implemented to run in time $O(nh)$, where $h$ is the complexity of the convex hull.

   e. What problems might occur when we deal with inexact floating point arithmetic?
4. Give an example of a simple polygon and a set of guards that sees all the “walls” of the polygon but yet misses some of the interior of the polygon.

5. Give an example of a simple polygon where allowing guards anywhere in the polygon requires fewer guards than restricting the guards to vertices only.