Assignment no. 1

due: November 13th, 2017

Many exercises (but not all) will be taken from the main textbook of the course *Computational Geometry Algorithms and Applications*[1] (CGAA). Some exercises are similar but not identical to exercises in the book. When a figure in the book could be helpful, the exercise number in the book is given.

PART I

Exercise 1.1 (a) Let P_1 and P_2 be two disjoint convex polygons in the plane with a total of n vertices. Give an $O(n)$-time algorithm to compute the convex hull of $P_1 \cup P_2$. (b) Based on (a), develop an $O(n \log n)$-time divide-and-conquer algorithm to compute the convex hull of a given set of n points in the plane.

Exercise 1.2 Devise an algorithm to compute the convex hull of a set of m possibly intersecting convex polygons in the plane, with a total of n vertices. Let h denote the number of vertices on the boundary of the desired convex hull. The algorithm should run in $O(mh + n)$ time—show that this is indeed the running time of your algorithm. Analyze the storage requirement of the algorithm.

Exercise 1.3 Describe in detail a "gift-wrapping" algorithm for computing the convex hull of a finite set of points in three-dimensional space and analyze its running time. You may assume that the input points are in general position, which means, in particular, that no four points lie on a common plane, that no three points lie on a common line, etc.

Exercise 1.4 Let S be a set of n disjoint triangles in the plane. We want to find a set of $n - 1$ segments with the following properties:

- Each segment connects a point on the boundary of one triangle to a point on the boundary of another triangle.
- The interiors of segments are pairwise disjoint and they are disjoint from the triangles
- Together they connect all the triangles to each other, that is, by walking along the segments and the triangle boundaries it must be possible to walk from a triangle to any other triangle.

Develop a plane sweep algorithm for this problem that runs in $O(n \log n)$ time. State the events and the data structures that you use explicitly, and describe the cases that arise and the actions required for each of them. Also state the sweep invariant. (CGAA Ex. 2.13)

Notice that the assignment continues on the other side of the page.

[1] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, 3rd Edition, Springer, 2008.

PART II

You may submit the following exercise one week later than Part I, by **November 20th, 2016**.

Exercise P1.1 Write a program to solve the following problem: Given an axis-parallel rectangle R and a simple polygon P in the plane, compute the area of the intersection of R with the convex hull of P. Here are possible steps towards a solution: (i) compute the area of the intersection of a triangle and an axis-parallel rectangle, (ii) decompose a convex polygon into triangles, and (iii) compute the convex hull of a simple polygon. You can use simple algorithms. The emphasis in this exercise is on the correctness of the result rather than on efficiency. However, try not to devise unnecessarily inefficient solutions, in particular the running time of your algorithm should not exceed $O(n^2)$, where n is the number of vertices of P. Also, you are required to write the **entire solution** from scratch and not use ready-made procedures. You may assume that the input polygon is a valid simple polygon, and that the input rectangle is a valid axis-parallel rectangle. The vertices of the polygon as well as of the rectangle will be given in integer coordinates in the range $[-10,000 : +10,000]$.

Feel free to write in C, C++, Java, or Python. Precise instructions on the I/O format as well as how your program should be activated will appear in the course's website.