Advanced Algorithms: Homework 6

Due on Mar. 15, 2024 at 11:59pm EST

Professor Dana Randall Spring 2024

As stated in the syllabus, unauthorized use of previous semester course materials is strictly prohibited in this course.

Exercise 1

Consider an online convex hull problem in which we are given the set s of n points one at a time. After receiving each point, we are to compute the convex hull of the points seen so far. Obviously we could run Graham's scan once for each point, with a total running time of $\mathcal{O}(n^2 \log n)$. Show how to solve the online convex hull problem in a total of $\mathcal{O}(n^2)$ time.

Exercise 2

Given n points in the plane, construct a simple (i.e., non-self-intersecting) polygon having these points as its vertices Devise an $\mathcal{O}(n \log n)$ algorithm and **show that this is optimal**.

Exercise 3

Given two arbitrary convex polygons P and Q with n vertices each, compute the convex hull of $P \cup Q$. Note that the two polygons can intersect any number of times, they can be disjoint, or one can be contained within the other.

(a) Specify as efficient an algorithm as you can to solve this problem, and analyze its complexity and prove its correctness.

(b) Use this algorithm to generate a divide-and-conquer algorithm without presorting that finds the convex hull of an arbitrary set of n points.

Exercise 4

Computational biologists often compute physical properties of large molecules. One of these properties is something called the *accessible surface area* of the molecule. We will consider this problem in a 2-dimensional setting.

We model the atoms of our molecule as a set of unit disks, each of radius of 1, centered at a given set of points $P = \{p_1, \ldots, p_n\}$. Let b_i denote the unit disk centered at point p_i. Let $M(P) = \bigcup_{i=1}^{n} b_i$ denote the union of these disks (shaded in blue in Fig. 1(a)). Let $\partial M(P)$ denote its boundary. Observe that $\partial M(P)$ is composed of circular arcs, and it may have multiple connected components (all of which contribute to the accessible surface).

(a) Present an $O(n \log n)$-time algorithm, which given a set P of n points in the plane, computes the length of accessible surface (see the solid curve in Fig. 1(a)).

Note: I am mostly interested in how to identify the circular arcs that make up the boundary of $M(P)$, not the actual perimeter value. To simplify matters, you may assume that you have access to a black-box function that computes the length of a given circular arc.

(b) In most applications in drug design, the molecule resides in liquid solvent, such as water. A statistic that is more relevant is something called the *solvent-accessible surface*. We model a water molecule as a disk of some radius r_0. Imagine that we roll such a disk around the entire boundary of $M(P)$ and trace out the path taken by the center of this disk (the dashed curve in Fig. 1(b)). As with the accessible surface, this boundary may have multiple components, all of which contribute to the final result.

Present an $O(n \log n)$-time algorithm, which given P and r_0, computes the length of the solvent-accessible surface (the dashed curve in Fig. 1(b)). *Hint:* This should be an easy extension to (a).

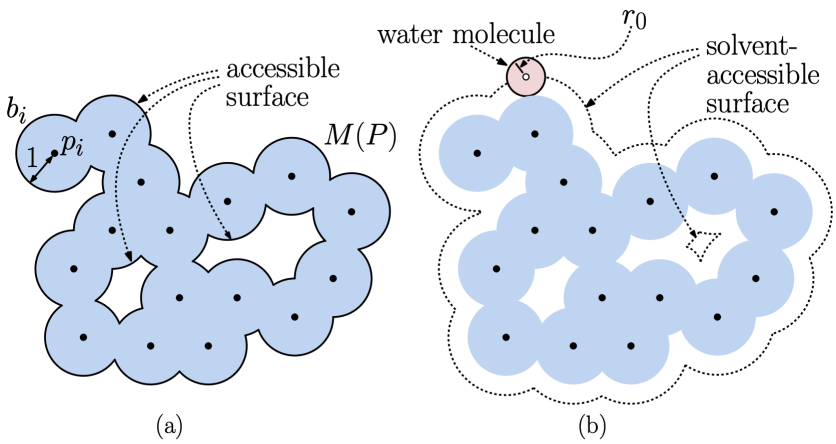

Figure 1: Solvent surface