1. Suppose that in Graham scan, pairs of points (but not three or more) may have the same x-coordinates. How can you modify the algorithm to handle this degeneracy.

2. [Incremental Convex Hull construction]

(a) Let C be a convex polygon, and P a point. Propose an algorithm to determine whether P is inside or outside C.

(b) Let $P_1, P_2, \ldots, P_n$ be a set of n points in the general position. We want to compute CH($P_1, P_2, \ldots, P_n$). Use Part (a) to convert CH($P_1, P_2, \ldots, P_i$) to CH($P_1, P_2, \ldots, P_{i+1}$). Total running time?

(c) Propose an O($n \log n$)-time algorithm using this idea.

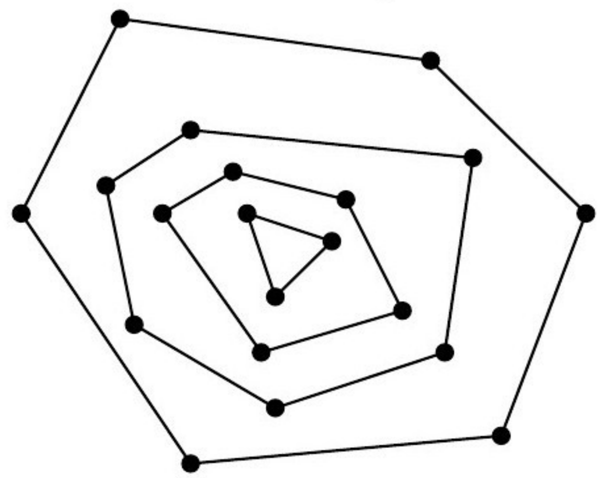
3. Let S and T be two disjoint sets of points in the Euclidean plane. S and T need not be horizontally separated. You have computed the two convex hulls CH(S) and CH(T).

(a) Propose an O(n)-time algorithm to merge these two hulls to CH($S \cup T$), where $n = | S \cup T |$.

(b) Prove that the problem of Part (a) cannot be solved in o(n) time in the worst case.

4. [Onion layers]

Let S be a set of n points in the plane. Let L_1 denote the set of vertices of CH(S). Remove the points of L_1 from S, and compute the convex hull of S again. Let L_2 denote the set of vertices of this convex hull. Repeat.



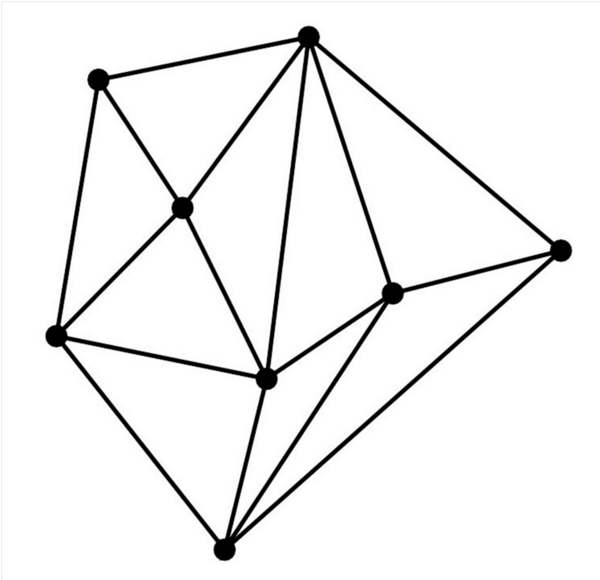
What is the worst-case running time for computing all onion layers if you use

(a) Jarvis march
(b) Graham scan.

Give tight bounds.

5. You are given n points in the plane in general position. Arrange the points to a list $P_1, P_2, \ldots, P_n$ such that each $P_iP_{i+1}P_{i+2}$ is a right turn.

6. (a) Prove that the number e of edges in any triangulation of CH(S) with $|S| = n$ satisfies $2n - 3 \leq e \leq 3n - 6$.



(b) How can you use the incremental convex-hull construction to triangulate CH(S)? Running time?

7. [Farthest Pair]

Let S be a set of n points in general position in the plane. We want to find P, $Q \in S$ such that $d(P, Q)$ is the maximum. This distance is called the diameter of S.

(a) Prove that P and Q are vertices of CH(S).

(b) Let P be a point on CH(S). Demonstrate that the distances of the points on CH(S) from P need not be unimodal.

(c) Let Q, Q' be consecutive vertices on CH(S). Let L be the line QQ'. The perpendicular distances of the vertices of CH(S) from L are unimodal. Let P be the farthest point from L. Build a collection C of pairs (P, Q) and (P, Q'). Prove that the farthest pair (P, Q) can be found in C.

(d) If the points in S are in general position, what is the maximum size of C?

(e) Propose an $O(n \log n)$-time algorithm for computing the farthest pair in S.

8. [A point sweep algorithm in one dimension]

(a) You are given n intervals $[a_i, b_i]$ standing for activities (like classes, seminars, and so on), all of which must be scheduled. Propose an efficient algorithm to find out the minimum number of classrooms needed. Assume that the interval endpoints are in general position.

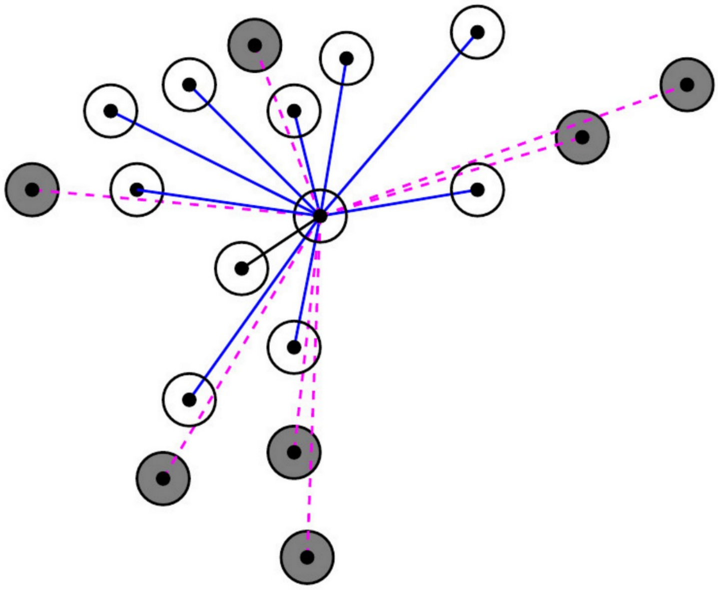
(b) Lift the general-position restriction from the previous exercise. How can you make your algorithm work (in the same running time)?

(c) Assume that each activity has a preparation time c_i and a closing time d_i. How can you make your algorithm work in this setting?

9. Suppose that in the line-sweep algorithm for the line-segment intersection problem, some lines are allowed to be vertical. Explain how you can handle a “vertical segment” event.

10.

There are n cell-phone towers in the plane. The towers are located at the points (x_i, y_i) for $i = 1, 2, 3, \ldots, n$. Around each tower, there is an interference zone of a fixed radius r (the same for all the towers). Assume that the interference zones do not overlap with each other. Two towers can communicate without interference if the line segment joining them does not intersect with the interference zone of any other tower. Your task is to determine all the towers with which the first tower (located at (x_1, y_1)) can communicate without interference. The following figure gives an example. Propose an $O(n \log n)$-time algorithm to solve this problem. Clearly mention the data structures that your algorithm uses. Also justify that your algorithm actually achieves the given running time.



11. Use a circle-sweep algorithm to solve Exercise 10. Here, a circle centered at Tower 1 grows from radius 0 to radius ∞.