Solution Sketch for Tutorial-3

1. Suppose you are given a matching M in a bipartite graph G = (V, E). Design a linear time algorithm to check if M is a maximum matching or not.

   [Solution sketch] Construct the flow network same as the one for computing maximum matching in bipartite graphs using max-flow done earlier, initialize the flow on edges corresponding to the matching (as well as edge from s and to t for the matched vertices) as 1. Now do one DFS/BFS to find an augmenting path in the residual graph. If an augmenting path is found, M is not a maximum matching, otherwise it is.

2. In the Stable Roommate problem, there is a set of 2n people, each of whom ranks everyone else in order of preference. The goal is to find a perfect matching (a disjoint set of n pairs) such that there is no unstable pair. Show that unlike the stable matching problem, there may not exist any stable matching in this case.

   [Solution sketch] Consider 4 persons, p1, p2, p3, p4, and the following preference list:

   p1: p2, p3, p4
   p2: p3, p1, p4
   p3: p1, p2, p4
   p4: anything

   Show that there is an unstable pair with one member = whoever is paired with p4.

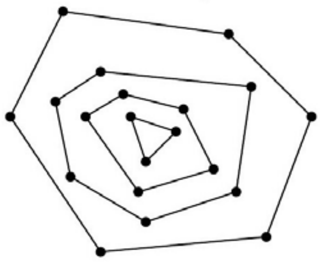
Solutions to the problems 3 to 7 below are included in AD's tutorial solution sent today. Look it up. Not shown again.

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

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

5. 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.

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

7. Triangulation of a convex polygon is a partition of the polygon into a set of triangles (so no two triangles have intersecting pairs, and the union of all the triangles is the polygon). How can you use the incremental convex-hull construction to triangulate the convex hull CH(S) of a set of points S such that all the points of the triangle are from S? Running time?