CSE 2012- Design and Analysis of Algorithms Practice Problem Sheet (Computational Geometry)

Practice makes you Perfect

1. Given the two end points P_1, P_2 of a line segement $\overline{P_1P_2}$, design an algorithm to generate n points on $\overline{P_1P_2}$. Your algorithm should get the input n and accordingly output the x-coordinate, y-coordinate of all the n points. Analyse your algorithm with the time complexity.
2. Given the two end points P_1, P_2 of a line segement $\overline{P_1P_2}$, and the coordinates of a point q, design an algorithm to check whether the point q in the clockwise direction from the line segment $\overline{P_1P_2}$ or not. Analyse your algorithm with the time complexity.
3. Given the coordinates of n points in a 2-dimensional plane, design an algorithm to identify the set of points which are collinear. There may be more than one set of points which are collinear. Your algorithm should print all the sets of points which are collinear. Analyse your algorithm with the time complexity.
4. Given the coordinates of the endpoints of two line segments $\overline{P_1P_2}$, $\overline{P_3P_4}$, design an algorithm to check whether the given line segments are parallel to each other or not. Analyse your algorithm with the time complexity.
5. Given the coordinates of the endpoints of two line segments $\overline{P_1P_2}$, $\overline{P_3P_4}$, design an algorithm to check whether the given line segments are perpendicular to each other or not. Analyse your algorithm with the time complexity.
6. Given the coordinates of the points $p_1, p_2, ..., p_n$ and an initial point p_0, design a pseudocode to sort the given n points in the increasing order of their polar angles with respect to the given origin point p_0. Your algorithm should sort in the increasing order of the angles without measuring the ploar angle of any point with reference to p_0. Analyse your algorithm with the time complexity.
7. Given the coordinates of three points A, B, C, design an algorithm to decide whether these vertices form a triangle or not. Analyse your algorithm with the time complexity.

8. Given the coordinates of the end points of n line segments, design an algorithm to retrun the line segments which intersect. If more than one pair of line-segments intersect, your algorithm should return all such pairs. Analyse your algorithm with the time complexity.

9. Given the coordinates of n points, design an algorithm to decide whether the given n points form a simple polygon or not. Analyse your algorithm with the time complexity.

10. Given the coordinates of the vertices $p_1, p_2, ..., p_n$ of a simple polygon P, design an algorithm to check whether the given polygon P is convex or not. Analyse your algorithm with the time complexity.

11. Given the coordinates of n points $p_1, p_2, ..., p_n$, design an efficient algorithm to identify the pair of points which are farthest from each other. Pair which are farthest are those pair whose distance from each other is maximum among the distances of all the pairs formed with the n given points. Your algorithm should not compute the distance between any two points for finding the farthest pair of points.

12. Given the coordinates of n points $p_1, p_2, ..., p_n$, design an efficient algorithm to identify the pair of points which are closest to each other. Pair which are closest are those pair whose distance from each other is minimum among the distances of all the pairs formed with the n given points. Your algorithm should not compute the distance between any two points for finding the closest pair of points.

13. The convex hull of a set Q of points is the smallest convex polygon P for which each point in Q is either on the boundary of P or in its interior. Design a divide-conquer-combine algorithm to compute the $CH(Q)$