CS31005 Algorithms – II

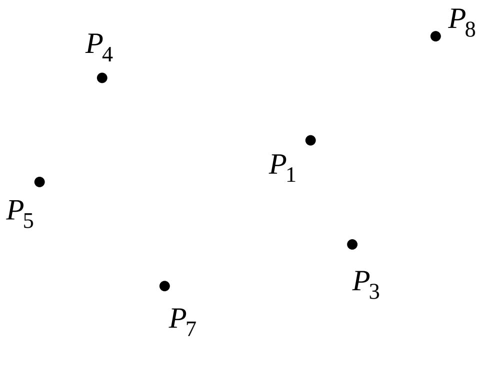
Class Test 2

Date: 02-Nov-2020 Maximum marks: 50

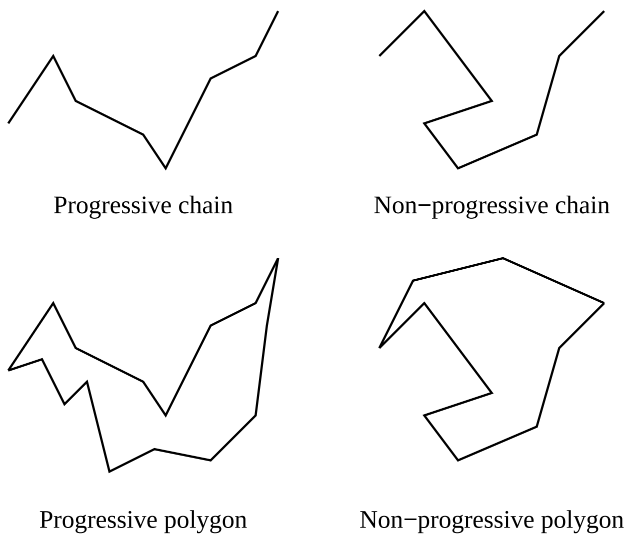
Instructions

- Answer all the questions. Be brief and precise.
- If you use any algorithm/result/formula covered in the lectures or the tutorials, just mention it, do not elaborate.

1. Explain the working of Graham's scan for computing the upper hull of the following set of eight points. **(10)**



2. A chain of non-intersecting line segments is called progressive if the x-coordinate always increases during the traversal of the chain from the leftmost point to the rightmost point. A simple (but not necessarily convex) polygon is called progressive if it consists of two progressive chains. The following figure demonstrates this concept.

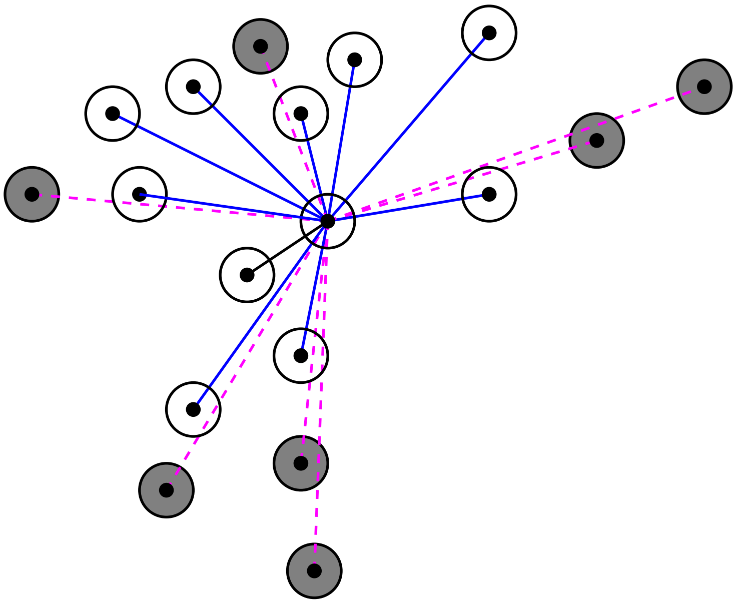


You are given a set S of $n \geqslant 3$ points. Your task is to construct a progressive polygon P having its only vertices at *all* of the given points in S. You may assume that the points in S are in general position.

(a) Prove/Disprove: For every S, the polygon P can be constructed uniquely. **(4)**

(b) Propose an $O(n \log n)$-time algorithm to construct P from S. **(6)**

3. 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. **(10)**



4. You are given two arrays $A = (a_1, a_2, a_3, \ldots, a_m)$ and $B = (b_1, b_2, b_3, \ldots, b_n)$ of *positive* integers. Your task is to find out whether there exist a *non-empty* subset I of $\{1, 2, 3, \ldots, m\}$ and a *non-empty* subset J of $\{1, 2, 3, \ldots, n\}$ such that $\sum_{i \in I} a_i = \sum_{j \in J} b_j$. Prove that this problem is NP-Complete. **(10)**

5. You are given an undirected graph $G = (V, E)$ and a positive integer k. You want to determine whether there exist k or fewer edges in E such that the removal of these edges from E makes G bipartite. Prove that this problem is NP-Complete. (**Hint:** Use the max-cut problem for the reduction.) **(10)**