CS 473: Fundamental Algorithms, Spring 2011

HW 2

Homework is due by **Monday, 23:59:59, February 7**

Problem 1 is due by **Sunday, 23:59:59, February 6**

This homework contains four problems. **Read the instructions for submitting homework on the course webpage**.

Collaboration Policy: For this homework, Problems 2–4 can be worked in groups of up to three students.

Problem 1 should be answered in Compass as part of the assessment HW2-Online and should be done individually.

---

1. (30 pts.) Short questions to be answered on compass individually.

2. (30 pts.)

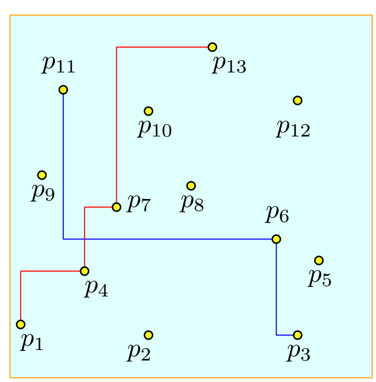
   (A) (15 pts.) Four cycles.
   Design and analyze an algorithm that takes as input an undirected graph $G = (V, E)$ and determines whether G contains a simple cycle (that is, a cycle which does not intersect itself) of length four. Its running time should be at most $O(|V|^3)$. You may assume that the input graph is represented either as an adjacency matrix or with adjacency lists, whichever makes your algorithm simpler.

   (B) (15 pts.) Shortest cycle.
   Give an algorithm that takes as input a directed graph with positive edge lengths, and returns the length of the shortest cycle in the graph (if the graph is acyclic, it should say so). Your algorithm should take time at most $O(|V|^3)$.

3. (20 pts.) Staircase to heaven, and back.

   You are given a set P of n points in the plane (i.e., each point is specified as a pair (x, y) of real numbers). A *staircase* is a sequence of points $q_1, q_2, \ldots, q_k$ such that for any $i < j < k$, we have that the point q_j is contained in the axis parallel rectangle having q_i and q_k as its corners. (Intuitively, a staircase can be going up or down.)

   As a concrete example, consider the figure on the right. Here the points p_1, p_4, p_7, p_{13} form a (upward) staircase. Similarly, p_{11}, p_6, p_3 form a (downward) staircase. (There are longer staircases in this point set.)

   Give an algorithm that computes the staircase using points of P, with the largest number of points of P in it. If your algorithm runs in $O(n^2)$ time, you will get full points.

   (Hint: The solution is fairly simple. As a first step, try to reduce this problem to a problem on a directed graph, for computing an upward staircase.)

4. (20 pts.) RELIABLE SHORTEST PATH.

You are given a directed graph G with positive weights on the edges, and a special source vertex s (G has n vertices and m edges). Think about this graph as a network (the weights represent transmission time on a connection). If a vertex v gets deleted (i.e., the router at this node failed) the distance between s and a vertex t in the remaining graph might increase substantially. Formally, let $G - v$ denote the graph G after v and all its edges get deleted. The *reliable shortest path distance* between s and t, denoted by $d_1(t)$ is the quantity $\max_{v \in V \setminus \{s,t\}} \mathrm{dist}_{G-v}(s,t)$, where $\mathrm{dist}_{G-v}(s,t)$ is the shortest path distance between s and t in $G - v$.

Provide an algorithm that computes $d_1(t)$ for all the vertices $t \in G$. How fast is your algorithm? The faster your algorithm is, the better it is.