CS/ECE 374 A (Spring 2022)
Homework 8 (due March 31 Thursday at 10am)

Instructions: As in previous homeworks.

Problem 8.1: We are given a set P of n points $\{p_1, \ldots, p_n\}$ in two dimensions. Each point p_i has coordinates (x_i, y_i). (In this question, the x_i's and y_i's may not be distinct!)

(a) Given a source point $s \in P$ and a destination point $t \in P$, we want to determine whether there exists a sequence $p_{i_0} p_{i_1} p_{i_2} \cdots p_{i_k}$ with $p_{i_0} = s$ and $p_{i_k} = t$, such that each segment $p_{i_j} p_{i_{j+1}}$ is either horizontal or vertical. (Paths are allowed to self-intersect.) Describe an efficient algorithm to solve this problem.

(b) Next, given a source $s \in P$ and a destination $t \in P$ and a value L, we want to determine whether there exists a sequence $p_{i_0} p_{i_1} p_{i_2} \cdots p_{i_k}$ with $p_{i_0} = s$ and $p_{i_k} = t$, such that (i) each segment $p_{i_j} p_{i_{j+1}}$ is either horizontal or vertical, and (ii) $d(p_{i_j}, p_{i_{j+1}}) + d(p_{i_{j+1}}, p_{i_{j+2}}) \geq L$ for all $j = 0, \ldots, k - 2$. (The motivation is in finding a path that avoids quick turns.) Describe an efficient algorithm to solve this problem.

You should solve the above problems by transforming the input into a graph and applying a standard algorithm you have seen in class. In these type of graph questions, remember to give mathematically precise definitions of the vertices and edges of your graph. Analyze the overall running time (the time to construct the graph plus the time to run the standard algorithm) as a function of n.

[Hint: For (a), partial credit will be given for an $O(n^2)$-time algorithm, but there is an $O(n \log n)$-time algorithm. For (b), construct a graph with $O(n^2)$ vertices...]

Below, the left picture shows one path that is a feasible solution for (a). The right picture shows a path that might work better for (b), depending on the value of L.
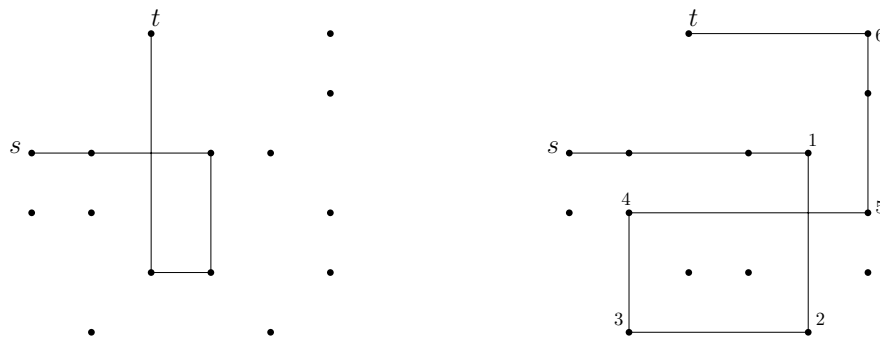
Problem 8.2: We are given a directed graph with n vertices and m edges and a source vertex s, where each edge e has a color $c(e)$ from $\{1, \ldots, k\}$.

  (a) (25 pts) Describe an algorithm to determine whether there is a closed walk that contains s and uses at least 4 colors. (Recall that in a *walk*, repeated vertices are allowed.) For full credit, the running time should be $O(m + n)$.

     [Hint: use the meta-graph.]

  (b) (75 pts) Describe an algorithm to determine whether there is a walk (not necessarily closed) that starts at s and uses at least 4 colors. For full credit, the running time should be $O(k^3(m + n))$ or better.

     [Hint: one approach is to define a new graph with $O(k^3 n)$ vertices. You will still get partial credit if you obtain $O(k^4(m + n))$ time instead.]