CS/ECE 374 A (Spring 2024)
Homework 8 (due Mar 28 Thursday at 10am)

Instructions: As in previous homeworks.

Note on Problem 8.1: you should solve each part by transforming the input into a graph and applying a standard algorithm you have seen in class: (i) give mathematically precise definitions of the vertices and edges of your graph (indicate if your graph is directed or undirected, weighted or unweighted); (ii) state which standard problem and standard algorithm you use on your graph (no need to redescribe the standard algorithm); (iii) analyze the overall running time (including the time to construct the graph, plus the time to run the standard algorithm) as a function of the original input size.

Problem 8.1: We are given a set S of n line segments (representing "roads"), each of which is either horizontal or vertical. A horizontal line segment may be specified by two x-coordinates and one y-coordinate. A vertical line segment may be specified by one x-coordinate and two y-coordinates. The segments may intersect each other (there could be $O(n^2)$ intersection points). (You may assume that all x- and y-coordinates are distinct.) We are also given a "start" point s lying on one of the line segments of S, and a "destination" point t lying on another line segment.

(a) (30 pts) Design an $O(n^2)$-time algorithm to find a path from s to t that stays on the input line segments in S, while minimizing the number of turns.

(Hint: construct a graph, where the vertices are just the line segments of S. Part (a) is meant to be easier than part (b).)

(b) (70 pts) Design an efficient algorithm to find a path from s to t that stays on the input line segments in S and *does not make two consecutive left turns*, while minimizing *the number of times the path passes through intersections.* Note that the path may self-intersect (and we may count an intersection point more than once).[1]

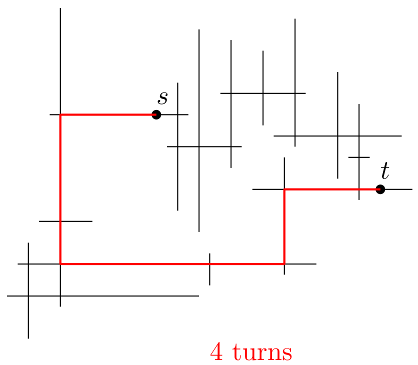
(Hint: construct a graph where the vertices correspond to the intersection points. You may also want to encode the previous two different directions that got us to the intersection point. In other words, try defining the vertex set to be

$$V = \{(p, D, D') : \quad p \text{ is an intersection point, and } (D, D') \in \{(\rightarrow,\uparrow),(\rightarrow,\downarrow),(\leftarrow,\uparrow),(\leftarrow,\downarrow),(\uparrow,\leftarrow),(\uparrow,\rightarrow),(\downarrow,\leftarrow),(\downarrow,\rightarrow)\}\}.$$

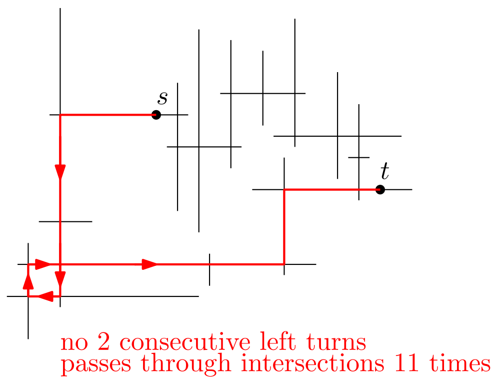
How should you define the edge set?

Note that an intersection point can be represented by a pair of horizontal and vertical line segment. For each intersection point p, it may be helpful to pre-compute the intersection point $\textsc{neighbor}_{\rightarrow}(p)$ immediately to the right of p, the intersection point $\textsc{neighbor}_{\leftarrow}(p)$ immediately to the left of p, the intersection point $\textsc{neighbor}_{\uparrow}(p)$ immediately above p, the intersection point $\textsc{neighbor}_{\downarrow}(p)$ immediately below p. How efficiently can you pre-compute these neighboring points?

For full credit, the total running time should be $O(n^2 \log n)$ or $O(n^2)$, although a correct $O(n^3)$-time algorithm would still receive up to 60 out of 70 pts.)



(a)


s
t
no 2 consecutive left turns
passes through intersections 11 times

(b)

Problem 8.2: We are given a directed graph $G = (V, E)$ with n vertices and m edges, where each edge is colored red or blue.

(a) (40 pts) For every vertex $v \in V$, we want to determine whether there exists a closed walk through v that uses at least one red edge. Design and analyze an efficient algorithm to solve this problem. For full credit, the running time should be $O(n + m)$.

(Hint: first compute the strongly connected components of the entire graph G… Remember to justify correctness of your algorithm.)

(b) (60 pts) For every vertex $v \in V$, we want to determine whether there exists a closed walk through v such that the number of edges on the walk is divisible by 374 (and is nonzero). Design and analyze an efficient algorithm to solve this problem. For full credit, the running time should be $O(n + m)$.

(Hint: first construct a new graph G' with $374n$ vertices, and compute the strongly connected components of G'… Remember to justify correctness.)

(c) (New, Bonus: 10 pts) For those who find part (b) too easy, here is a bonus problem: For every vertex $v \in V$, we want to determine whether there exists a closed walk through v such that the number of red edges on the walk is congruent to 1 mod 374. Design and analyze an efficient algorithm to solve this problem. To get the 10 extra pts, the running time should be $O(n + m)$, and you need to give valid justification of correctness.

[1]Note: we are **not** allowed to make 180-degree "U-turns" at an intersection.