CS/ECE 374 A (Spring 2024)
Homework 9 (due Apr 4 Thursday at 10am)

Instructions: As in previous homeworks.

Problem 9.1: We are given a weighted DAG (directed acyclic graph) $G = (V, E)$ with n vertices and m edges ($m \geq n$), where each vertex v has a number $c(v)$. We are also given vertices $s, t, s', t' \in V$, an integer $k \leq n$, and a number L. We want to find two length-k paths $\langle s, u_1, u_2, \ldots, u_{k-1}, t\rangle$ and $\langle s', v_1, v_2, \ldots, v_{k-1}, t'\rangle$ in G, such that $\sum_{i=1}^{k-1} |c(u_i) - c(v_i)| \geq L$.

Describe an efficient algorithm to solve this problem. Do not use dynamic programming. Instead construct a new graph G' (hint: use $O(kn^2)$ vertices) and run some known shortest-path algorithm on this graph. Analyze the running time as a function of m, n, k.

Problem 9.2: We are given a weighted directed graph G with n vertices and m edges ($m \geq n$), where all edge weights are *positive* and each edge is colored red or blue.

(a) (30 pts) Describe an efficient algorithm to determine whether there exists a cycle in G such that strictly more than 1/3 of the edges are red.

Hint: Just apply a known algorithm. Observe that the condition is the same as: (# blue edges) $- 2 \cdot$ (# red edges) < 0.

(b) (70 pts) Describe an efficient algorithm to find a cycle in G such that strictly more than 1/3 of the edges are red, while minimizing the sum of the edge weights in the cycle. For full credit, the running time should be close to $O(mn^2)$ (possibly with some logarithmic factors).

Hint: construct a new graph G' where a vertex is of the form (v, i) where $v \in V$ and i is a (positive or negative) number in a certain range...

Note: the minimum-weight closed walk satisfying the condition must be a simple cycle, but why? This is needed to justify correctness.