HW 10

Due on **Wednesday, November 16, 2022 at 9am CST**

CS/ECE 374A: Intro. Algorithms & Models of Computation, Fall 2022 Version: **1.0**

Submission instructions as in previous homeworks.

19 (100 PTS.) The best path.

Let $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ be a directed graph with n vertices and $m \geq n$ edges, and distinct positive real weights on the edges (here $w(e)$ denotes the weight of an edge $e \in \mathsf{E}$). For two sets X, Y let $X \oplus Y = (X \setminus Y) \cup (Y \setminus X)$ be their symmetric difference. For two paths π, σ in G, let e be the most expensive edge in $\mathsf{E}(\pi) \oplus \mathsf{E}(\sigma)$. If $e \in \pi$, then we write $\pi \succ \sigma$ (i.e., π is ***worse*** then σ). Clearly, this defines a natural ordering on the paths in G. The ***best*** path between s and t in G, is the unique path, such that all other paths (from s to t) are worse than it. Informally, the best path between s and t is the path minimizing the maximum weight edge on the path, and this property holds recursively on the two subpaths after we remove this edge.

Given vertices s and t, describe an algorithm, as fast as possible, that computes the best path from s to t.

Prove ***formally*** that the path your algorithm output is indeed the best path.

Partial credit would be given to efficient suboptimal algorithms.

(Hint: Think about the algorithm for the problem for the undirected case.)

20 (100 PTS.) Downwind from minus infinity.

Let G be a directed graph with n vertices and m edges, with distinct real weights (denoted by $w(\cdot)$) on the edges. A vertex $v \in \mathsf{V}(\mathsf{G})$ is ***sad*** if for any real number $\beta < 0$, there exists a walk π in G that ends in v, and $w(\pi) = \sum_{e \in \pi} w(e) < \beta$. Describe an algorithm (using or modifying algorithms seen in class), as fast as possible, that computes *all* the sad vertices of G.