Homework Assignment #8
Due: March 28, 2025 at 5:00 p.m.

The same rules apply as for Assignment 1. (In particular, you can work in pairs, where each pair submits just one paper.)

1. Given a directed graph G with non-negative weights on the edges, we wish to find shortest paths that contain exactly k edges. The paths are allowed to contain loops (i.e., a path may visit the same node more than once). If an edge is used multiple times in a path, each use of it counts towards k. Assume the nodes are numbered $1..n$ and that there are m edges, with $m > n$. Let $w(i, j)$ denote the weight of the edge from i to j. We shall consider two approaches in parts (b) and (c).

[1] **(a)** Give an example of a graph where the shortest path from some node s to some other node t that contains exactly 5 edges includes a loop.

[4] **(b)** Give a precise description of how to construct a new graph G' with nk nodes so that running the standard version of Dijkstra's algorithm on G' allows you to find the shortest paths in G that contain exactly k edges (from all nodes to a single destination node).
Give a good bound on the running time for Dijkstra's algorithm on G' in terms of n, m and k.

Hint: think of G' as having k layers of nodes, where edges go between layers.

[6] **(c)** Let D_ℓ be an $n \times n$ array, where $D_\ell[i, j]$ is the length of the shortest path from i to j with exactly ℓ edges (or ∞ if no such path exists).

- What is $D_0[i, j]$?
- If you have the array $D_{\ell-1}$, how can you compute the entries of D_ℓ? Briefly justify your answer.
- If ℓ is even and you have the array $D_{\ell/2}$, how can you compute the entries of D_ℓ? Briefly justify your answer.
- Describe how to compute D_k in $O(n^3 \log k)$ time.

[1] **(d)** Suppose you want to compute the shortest path containing exactly k edges from one node s to one other node t. Is the algorithm of part (c) ever preferable to the algorithm of part (b)?