Question 5 by Dan Nguyen (z5206032)

There is a given directed graph, $G(V, E)$, where each edge, e, has a weight, $0 < w_e < 1$.

The *safety* of a path from u to v with edges $e_1, e_2, ..., e_k$ is $\prod_{i=1}^{k} w_{e_i}$.

Let there be a matrix, D, which is initialised by G's adjacency matrix representation.

Consider a modified Floyd-Warshall algorithm which stores the *safety* from u to v in D instead of the distance, and the maximum *safety* is preferred i.e. the computed *safety* replaces the original *safety* if the computed value is greater than the original.

Define $Q(i, j, k)$ as the problem of finding the *safety* from a node i to j for $G[1..k]$ ending with $G[k]$.

Define $\text{opt}(i, j, k)$ as the solution to $Q(i, j, k)$.

For all $1 \leq i,\ j \leq n$, and $0 \leq k \leq n$, solve for $Q(i, j, k)$ using dynamic programming where the recurrence is:

$$\text{opt}(i, j, k) = \max\{\text{opt}(i, j, k-1),\ \text{opt}(i, k, k-1) + \text{opt}(k, j, k-1)\}$$

The base case is:

$$\text{opt}(i, j, 0) = \begin{cases} 1 & \text{if } i = j \\ w(i, j) & \text{if } (i, j) \in E \\ 1 & \text{otherwise} \end{cases}$$

The order of solving Q is important i.e. subproblems with lesser i and j are solved first.

The final answer is:

$$\text{opt}(n, n, k) = \max\{\text{opt}(n, n, k-1),\ \text{opt}(n, k, k-1) + \text{opt}(k, n, k-1)\}$$

The Floyd-Warshall algorithm has a worst time complexity of $O(|V^3|) = O(n^3)$ for G with n vertices.

For a set of ordered pairs of vertices, $S = \{(u_1, v_1), ...\}$, S has a capacity of n^2 pair combinations since there are n vertices in G. Looking up $D(u, v)$ for the maximum *safety* from u to v for at worst n^2 pair combinations has a time complexity of $O(n^2)$.

The overall worst time complexity is $O(n^3) + O(n^2) = O(n^3)$ as required.