Q4

Subproblem

For start vertex S and destination vertex X, $S, X \in V$, and length j, $j \in [1, k]$. We define subproblem $P(S, X, j)$ to be "find the longest length from vertex S to vertex X through j edges"

Let $dp[S][X][j]$ be the longest path(i.e. total weight is maximum) from node S to node X using exactly J edges in total, which is the optimal solution to the subproblem $P(S, X, j)$.

Build-up order

For every vertex $S \in V$, and for every vertex $X \in V$, also for length $j \in [1, k]$. We handle subproblems in the order $P(S_1, X_1, 1), \ldots, P(S_1, X_1, k), P(S_1, X_2, 1), \ldots, P(S_1, X_2, k), \ldots, P(S_n, X_n, 1), \ldots, P(S_n, X_n, k)$.

We can use nested loops to traverse all vertices as start node and destination node.

Base case

For vertex $S \in V$,

$$dp[S][S][0] = 0$$

For vertices S and X, $S, X \in V$, and length $j = 1$, if S and X are connected,

$$dp[S][X][1] = weight(S, X)$$

Otherwise we don't store $dp[S][X][j]$.

Recursion

For start vertex S, destination vertex X, $S, X \in V$ and length $j \in [2, k]$, we have,

$$dp[S][X][j] = max\{dp[S][Y][j-1] + weight(Y, X)\}$$

for all Y which has an edge from Y to X and $Y \in V$.

Final solution

The final solution of the problem is the maximum value returned by any of these subproblems,

$$max\{dp[S][X][k] : S, X \in V\}$$

Time complexity

The complexity is $O(V^2 * E * k)$ because for two vertices, start vertex and destination vertex, which means one subproblem, the algorithm runs in $O(E * k)$, and there are V^2 subproblems, so the total runs in $O(V^2 * E * k)$.