COMP3121 Assignment 3 - Q5

Demiao Chen z5289988

July 27, 2021

Answer

We first initialise a 2D array with size $|V|$ x $|V|$, weight[$|V|$][$|V|$], to store the weights between two vertices in $G(V, E)$. The diagonal entries in weight[$|V|$][$|V|$] will be 0, as visiting a node from itself will have no weight. Nodes not directly reachable will be infinity.
Second initialise a 3D array of size $|V|$ x $|V|$ x K, dp[s][d][k], to represent the maximum total wight path from node s to d using exactly k edges in total. Initialise the whole array to infinity except for $d = s$, which sets to 0.
We now start finding the the maximum total weight path. For each s in V (V is the set of vertex in graph G), we loop K times, in each loop we examine every d in V, and update dp by do the following:

$$dp[s][d][k+1] = max(arr[s][i][k] + weight[i][d])$$

for all i which has an edge from i to d, i.e. $weight[i][d]$ is not infinity or 0.
Here $arr[s][i][k]$ is a copy of $dp[s][d][k]$ before updating dp.
After finishing above, we loop to find the maximum number in dp[s][d][K], which is the the maximum total weight of exactly length K we can get from G.
To get path, we use a recursive function and an array path[] to store the path,

$$opt(s, d, k) = max\{opt(s, i, k) + weight[i][d] : \text{weight[i][d] is not infinity or 0}\}$$

We add all i chosen from max{} to array path, where i represents each node visited, and reverse path when finishing the whole recursive process, then we get the maximum total weight of exactly length K path from the array path.
Time complexity is $O(K * |V|^2)$, as we have $|V|$ vertices in G, we have chosen every vertex as source node s, and for each source node s we loop K times to check every destination node d. Hence the time complexity is $O(|V| * K * |V|) = O(K * |V|^2)$.