Dheeraj Satya Sushant Viswanadham
(z5204820)

COMP3121 Assignment 4 – Question 4

4) We are required to find a path G of length K that contains the maximum total weight in the weighted directed graph G(V, E). We are also told that path in G can be possibly self-intersecting and can visit a vertex multiple times and traverse an edge multiple times. It can start and end at arbitrary vertices or start and end at the same vertex. Also, the edge between adjacent vertices is always unit in length but can have different weights placed on each edge and the weights will always be positive. We can also assume that there will also be a path of length K in our graph G.

Note that this is very similar to the Floyd-Warshall algorithm where instead of trying to find the least weighted path, we are instead trying to find the path with the maximum total weight in the weighted directed graph. Additionally, once we have tweaked our Floyd-Warshall algorithm, we can then specify our length K (which will be an additional variable to be added to the algorithm) and find the maximum total weight possible with our given K length.

We can try to obtain the maximum weighted graph by attempting to determine the maximum weight path of exactly length k which ends at node i where $1 \leq k \leq K$. Since each edge has positive edge weights, we can modify the algorithm to find the max weight from a node *i* to a node j for a given length k and have to solve subproblems of the form "find the maximum weight path of length k ending at node i". Let us consider a subset X = {1, 2, … , n} of vertices for some k. For any pair of vertices $i, j \epsilon V$, we need to find all paths from *i* to *j* whose intermediate vertices are from our given subset, and let *w* be our **maximum-weight path** among them. We will have our Floyd-Warshall algorithm exploit the relationship between path *w* and k-length paths from i to *j* with all intermediate vertices in the set {1, 2, … , *n* – 1} and this relationship depends on whether or not *n* is an intermediate vertex of path *w*.

If our *n* is not an intermediate vertex of path *w* and length *k*, then all intermediate vertices of path *w* and length *k* are in the set Y = {1, 2, … , n – 1}. As a result, the maximum-weight path of length k from node *i* to node *j* with all intermediate vertices in Y is also the maximum-weight path of length *k* from *i* to *j* with all intermediate vertices in X.

If our *n* is an intermediate vertex that lies on our path *w* of length *k*, then we can breakdown our *w* into w_1 from *i* to *n* and w_2 from *n* to *j*. As a result, our w_1 is the maximum-weight path from *i* to *n* with all intermediate vertices in X and by association, all intermediate vertices of w_1 are in Y. Similarly, w_2 will be our maximum-weight path from *n* to *j* with all intermediate vertices in Y. We can then use the bottom-up approach defined in the Floyd-Warshall algorithm (with the extra variable for length K), to find the maximum-weight path of length K in G(V, E). Since, we will have an extra nested for loop to account for the extra variable of length K when filling in our table and calculating the maximum-weight path from node *i* to node *j*, we will have an overall time complexity of $O(n^4)$.

Sources: CLRS (textbook) and lecture notes regarding the above modified Floyd-Warshall algorithm.

End of Solution