COMP3121 Homework Q4

Arth Sanskar Patel
z5228942

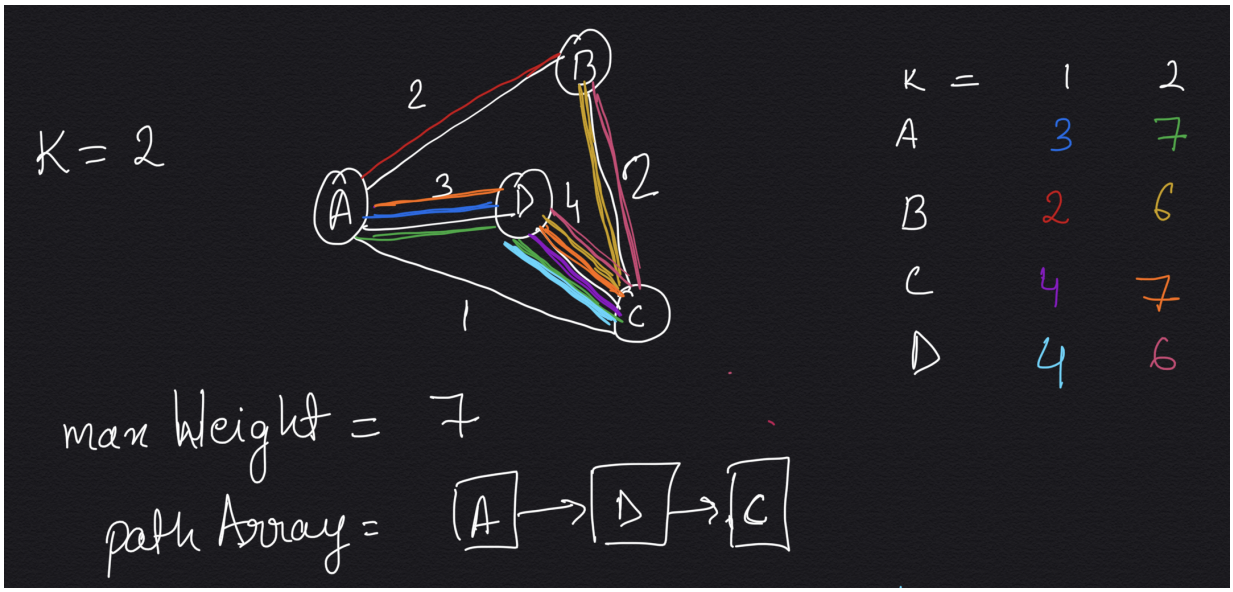
July 30, 2020

1 Answer

Let there be a graph G(V, E) with number of vertices nV and number of edges nE. We need to find a path of length exactly K which as highest weight possible. Here the path can go through a vertex and edge multiple times. Lets initialise an pathArray array of length K which will store the path and a maxWeigth variable of value $-\infty$. We then run a for loop for each k where $1 \leq k \leq K$, basically from 1 to K. Now inside this for each vertex i in V, we find a path of length k which has the maximum weight and ends on i. We record this path in pathArray and change the value of maxWeight if the weight of this path is greater than the current max weight. This way in the end we will be left with a path in our pathArray and the maximum weight will be the variable maxWeight. Pseudo code for this algorithm is given below:

```
maxWeight = -(inf)
pathArray = array(size(K))
Graph G = G(V, E)
# K + 1 because we need to include K.
for k in range(1,K+1):
    for i in range(0,nV):
        sourceVertex = G[i]
        for j in rest of vertices:
            pathArrayTemp = getPath(j, sourceVertex)
            if len(pathArrayTemp) == k and weight(pathArrayTemp) > maxWeight:
                pathArray = pathArrayTemp
                maxWeight = weight(pathArrayTemp)
```

Lets take an example to see how the algorithm works. Consider the graph and table below: The table on the right



shows the progression of our for loops with each color associated with the path that got the weight in each case. For example, the yellow path from $D- > C- > B$ corresponds to yellow 6 in the in the table at (B, 2), it is basically the weight of path of length 2 with the vertex ending at B. So after ending all the iterations, we will get the path with highest weight. This here is a non directional graph for simpler explanation purposes. Since this algorithm is an adaptation of Floyd-Marshall Algorithm for finding shortest path, it can be applied to directional graphs as well. The getPath function in the pseudo code will have a if loop which checks if there's an edge from A to B like any other get path algorithm such as DFS or BFS. Reference from this was taken from wikipedia for Floyd-Marshall Algorithm.