CX2101 Algorithm Design and Analysis

Tutorial 2: Graphs

School of Computer Science and Engineering *Nanyang Technological University*

Week 6 (Q1-Q3):

Q1 Apply the Dijkstra's algorithm on the graph represented by the following adjacency matrix to find the shortest distances and the shortest paths from vertex 1 to the other vertices. Show the contents of arrays S, d and pi after each iteration of the while loop.

$$\begin{array}{c c} \text{vertex} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} & \begin{bmatrix} 0 & 4 & 2 & 6 & 8 \\ \infty & 0 & \infty & 4 & 3 \\ \infty & \infty & 0 & 1 & \infty \\ \infty & 1 & \infty & 0 & 3 \\ \infty & \infty & \infty & \infty & 0 \end{bmatrix} \end{array}$$

Q2 Let $G = (V, E, W)$ be a weighted graph, and let s and z be distinct vertices. In the graph, there may be more than one shortest path from s to z. Explain how to modify Dijkstra's shortest-path algorithm to determine the number of distinct shortest paths from s to z. Assume all edge weights are positive.

Q3 Dijkstra's algorithm requires that the input graph has all edges being non-negative. Give an example where Dijkstra's algorithm does not work correctly with negative weights.

Week 7 (Q4-Q6):

Q4 Execute by hand the Prim's algorithm for finding minimum spanning tree (MST) on the graph in Figure 2.1, starting from vertex G. Show the contents of arrays S, d and pi after each iteration of the while loop when a vertex is added to the MST.

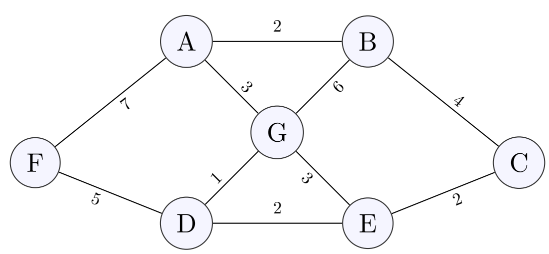

Figure 2.1: Graph for Q4

Q5 In a weighted undirected graph, is the path between two vertices in a minimum spanning tree always the shortest path (i.e. a path with the minimum weight) between the two vertices in the graph? If your answer is yes, give a proof; otherwise, give a counterexample.

Q6 Draw a connected graph with five nodes, six edges of respective weights 5, 6, 7, 8, 9, 10, and a minimum spanning tree of weight 28. Is it possible to have an MST of weight 29? If yes, draw the graph; otherwise, provide your justification.

Week 8 (Q7-Q9):

Q7 Execute by hand the Kruskal's algorithm (with the weighted QuickUnion algorithm for Union-Find) for finding minimum spanning tree (MST) on the graph in Figure 2.1. Show the contents of arrays id and sz at each step when an edge is added to the MST.

Q8 If the input graph to the Kruskal's algorithm is given in an adjacency matrix, what is the time complexity of the algorithm?

Q9 Design an algorithm to check whether a given undirected graph $G = (V, E)$ contains a cycle or not. Analyze the complexity of the algorithm in terms of $|V|$ and $|E|$.