The LNM Institute of Information Technology

Design & Analysis Of Algorithms

Assignment 3

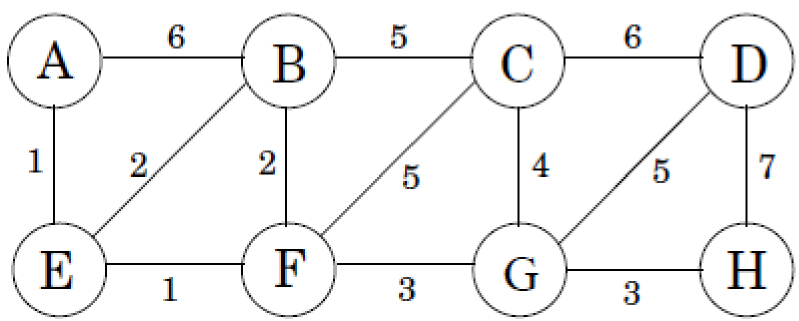
Q1. Rewrite the DFS procedure, using stack to eliminate the recursion.

Q2. The **square** of a directed graph G=(V,E) is the graph G^2 =(V,E^2) such that (u,v) ∊ E^2 if and only G contains a path with at most two edges between u and v. Describe efficient algorithms for computing G^2 from G using the adjacency-matrix representations of G. Analyze the running time of your algorithm.

Q3. The **diameter** of a tree T = (V,E) is defined as $max_{\boldsymbol{u,v \in V}}\ \delta(u,v)$, that is, the largest of all shortest-path distances in the tree. Give an efficient algorithm to compute the diameter of a tree, and analyze the running time of your algorithm..

Q4. Another way to perform topological sorting on a directed acyclic graph G=(V,E) is to repeatedly find a vertex of in-degree 0, output it, and remove it and all of its outgoing edges from the graph. Explain how to implement this idea so that it runs in time O(V+E). What happens to this algorithm if G has cycles?

Q5. Consider the following graph



(a) What is the cost of its minimum spanning tree?
(b) How many minimum spanning trees does it have?
(c) Suppose Kruskal's algorithm is run on this graph. In what order are the edges added to the MST? For each edge in this sequence, give a cut that justi_es its addition.

Q6. Consider an undirected graph $G = (V, E)$ with nonnegative edge weights w_e Suppose that you have computed a minimum spanning tree of G, and that you have also computed shortest paths to all nodes from a particular node $s \in V$.
Now suppose each edge weight is increased by 1: the new weights are
$w'_e = w_e + 1$.
(a) Does the minimum spanning tree change? Give an example where it changes or prove it cannot change.
(b) Do the shortest paths change? Give an example where they change or prove they cannot change.

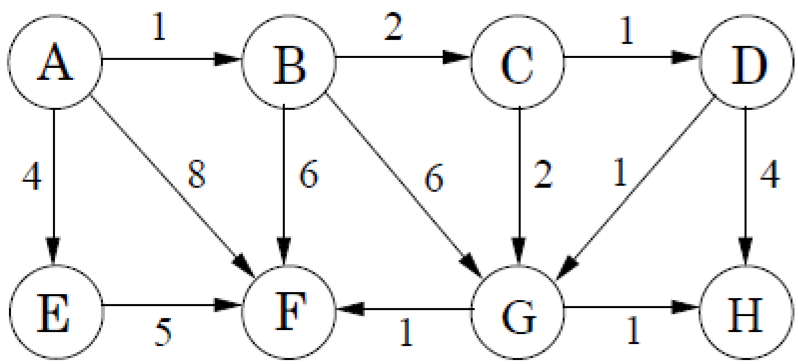
Q7. Given a graph G and a minimum spanning tree T , suppose that we decrease the weight of one of the edges not in T . Give an algorithm for finding the minimum spanning tree in the modified graph.

Q8. Suppose that all edge weights in a graph are integers in the range from 1 to |V|. How fast can you make Kruskal's algorithm run? What if the edge weights are integers in the range from 1 to W for some constant W ?

Q9. We use Huffman's algorithm to obtain an encoding of alphabet $\{a, b, c\}$ with frequencies f_a, f_b, f_c. In each of the following cases, either give an example of frequencies (f_a, f_b, f_c) that would yield the specified code, or explain why the code cannot possibly be obtained (no matter what the
frequencies are).
(a) Code: {0, 10, 11}
(b) Code: {0, 1, 00}
(c) Code: {10,01, 00}

Q10. You are given a directed graph with (possibly negative) weighted edges, in which the shortest path between any two vertices is guaranteed to have at most k edges. Give an algorithm that finds the shortest path between two vertices u and v in O(k|E|) time.

Q11. Suppose Dijkstra's algorithm is run on the following graph, starting at node A.



(a) Draw a table showing the intermediate distance values of all the nodes at each iteration of the algorithm.
(b) Show the final shortest-path tree.

Q12. Just like the previous problem, but this time with the Bellman-Ford algorithm.