DAA Tutorial-6
Graph Algorithms

Problem 1

(a) Traverse an unweighted graph using BFS and DFS, and list the order of vertex visits.

(b) Using DFS, record discovery and finishing times, and classify edges as tree, back, forward, or cross.

(c) Extend DFS to detect cycles in both directed and undirected graphs.

Problem 2

(a) Determine a valid ordering of vertices for a directed acyclic graph.

(b) Identify all strongly connected components of a directed graph and list the vertices belonging to each component.

Problem 3

(a) Construct a minimum spanning tree by repeatedly selecting edges in increasing order of weight. Show the edges selected and the total weight of the tree.

(b) Construct a minimum spanning tree by expanding from a starting vertex and adding the lightest edge connecting the tree to a new vertex at each step. Compare the resulting tree with part (a).

Problem 4

(a) Determine shortest paths from a single source in a graph with non-negative edge weights. Reconstruct the shortest path to a specified target vertex.

(b) Determine shortest paths from a single source when negative edge weights may be present. If a negative-weight cycle is reachable from the source, detect and report it.

(c) Compute shortest paths between all pairs of vertices and detect the presence of any negative-weight cycles using the distance matrix.

Problem 5

(a) Compute the maximum possible flow from a designated source to a sink in a directed network with edge capacities. Report:

- The value of the maximum flow.
- Augmenting paths and their bottleneck capacities.
- The final residual network.
- A minimum cut separating the source and sink, along with its capacity.

(b) Model a bipartite matching problem as a flow network and compute the maximum matching. Report the matching size and the matched pairs.

(c) Compute a maximum flow of minimum total cost in a capacitated, cost-weighted network and report the flow value and cost.