Assignment 2

Marks: 40 Due: December 7, 2025

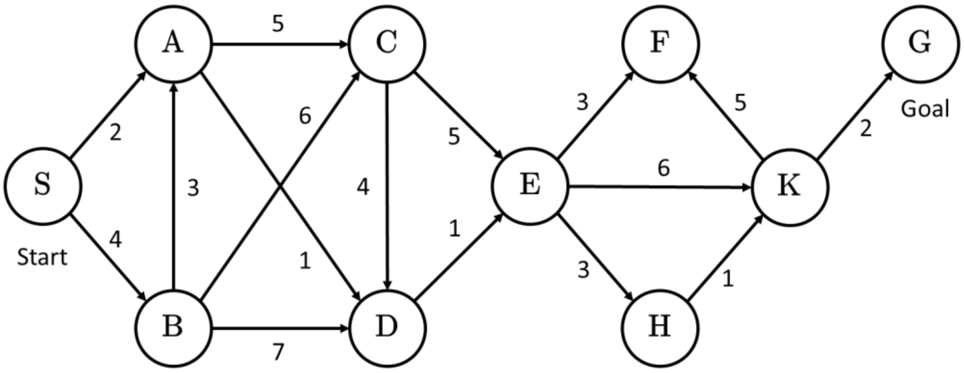
Question 1: Dynamic Programming (10 marks)

The Longest Increasing Subsequence (LIS) problem is a classic computer science challenge that requires finding the longest possible subsequence of a given sequence of numbers such that the elements in the subsequence are in strictly increasing order.

Given an input array [10, 9, 2, 5, 3, 7, 101, 18],
An increasing subsequence is [2, 3, 7, 18] or [2, 5, 7, 101].
The Longest Increasing Subsequence has a length of 4.

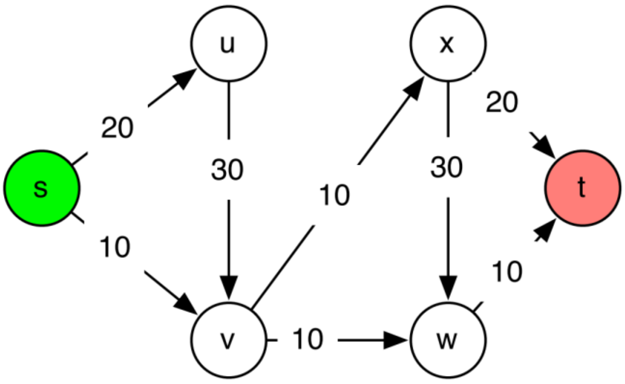
a) Provide a step-by-step explanation of how the LIS is determined in a given sequence of numbers [10, 9, 2, 5, 3, 7, 101, 18] using dynamic programming. (5)
b) Implement the Longest Increasing Subsequence (LIS) problem in **Python**. Provide the Python code for solving LIS and explain its time & space complexity. (5)

Question 2: Graph Algorithms (10 marks)



Given the directed weighted graph, apply Dijkstra’s Algorithm to find the shortest path from the start vertex to the goal vertex in the graph. (10)

Question 3: Network Flow (10 marks)



a) Consider the graph for finding the maximum flow in the network with capacities between nodes. Use the Ford-Fulkerson algorithm to solve the problem. (8)

b) Discuss the efficiency of the Ford-Fulkerson algorithm. What is its time complexity in the case of using BFS as the augmenting path search method? (2)

Question 4: NP-Completeness (10 marks)

a) Explain the Hamiltonian Cycle problem. Then, prove that the Hamiltonian Cycle problem is **NP-complete** by reducing it from the 3-SAT problem. (5)

b) Discuss the significance of NP-completeness in real-world problems. How does the concept of NP-hard problems impact algorithmic research and practical applications? (5)