Assignment 04

A practical study of graph algorithms

Your Name
Data Structures — Spring 2026

Due 18 March 2026

Submission checklist. Show enough work to make your reasoning verifiable. State the running time of every algorithm and cite any external source that informed your solution.

Problem 1 — Breadth-first search

Let $G = (V, E)$ be an undirected graph with $|V| = n$ vertices and $|E| = m$ edges. Describe an algorithm that returns the shortest unweighted distance from a source s to every reachable vertex. Give a loop invariant for the queue-processing loop and prove its running time.

Solution. Place s in a FIFO queue with distance zero. Whenever a vertex is removed, inspect its adjacency list and enqueue each undiscovered neighbour with distance one greater than the current vertex. The invariant is that a vertex leaves the queue only after its stored distance is its shortest path length: the first path reaching it has the smallest possible number of edges because the queue processes vertices in nondecreasing distance order. Each vertex and edge is inspected at most once, so the time is $O(n + m)$ and the storage is $O(n)$.

Problem 2 — A weighted counterexample

Give a small weighted graph showing why breadth-first search cannot replace Dijkstra's algorithm when edge weights are positive but unequal. Explain the incorrect choice it makes.

Solution. Use vertices s, a, b, t and edges

$$s - a \ (1), \qquad a - t \ (10), \qquad s - b \ (4), \qquad b - t \ (1).$$

The two-edge path through a is discovered first, but costs 11; the path through b costs 5. Breadth-first search minimises the number of edges, not their total weight.

Problem 3 — Design question

A graph is stored as an adjacency matrix. What is the asymptotic cost of testing whether an edge (u, v) exists, and what is the cost of enumerating all neighbours of u? Contrast both answers with an adjacency-list graph.

Claim. *For sparse graphs, adjacency lists use less space and make neighbour enumeration proportional to the degree of the selected vertex [1].*

References

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 3 edition, 2009.