# AM5801: Computational Lab
# Assignment 6

Date: September 25, 2025

Deadline: October 1, 2025 Max mark: 50

---

1. The metro rail authority of a metropolitan city is developing a smart navigation system to assist passengers in planning their journeys. The entire metro network is modeled as an undirected, unweighted graph $G = (V, E)$, where each station is represented as a vertex and each direct railway connection between two stations is represented as an edge. Given the total number of stations $N$, the number of direct railway connections $M$, followed by $M$ pairs of integers $(u, v)$ representing the connections, and finally two integers $s, t$ denoting the source station and the destination station, the objective is to design and implement an algorithm based on Breadth-First Search (BFS) that can efficiently guide passengers. For every station $v \in V$, the system should calculate the minimum number of stops required to travel from the source station $s$ to station $v$. If a station is unreachable from $s$, its distance should be reported as $\infty$. Additionally, for the specific destination station $t$, the system should print one shortest travel route from $s$ to $t$ as a sequence of stations. If multiple shortest routes exist, the system should also report the total number of distinct shortest routes so that passengers have the option of alternative paths in case of service interruptions. The program must accept the input values $N, M, (u, v)$ pairs, and $s, t$, and produce as output the minimum number of stops to each station, the sequence of stations along one shortest path from $s$ to $t$, and the number of distinct shortest routes from $s$ to $t$.

   The solution must run in $O(N + M)$ time, be capable of handling up to $10^5$ stations and $2 \times 10^5$ connections efficiently, and produce outputs in a clear and consistent format. For example, in a network with 6 stations and direct connections $(0, 1), (0, 2), (1, 3), (2, 3), (3, 4), (4, 5), (2, 5)$, with source $s = 0$ and destination $t = 5$, the algorithm should output the minimum stops to all stations, the shortest path from 0 to 5 as $0 \to 2 \to 5$, and the number of distinct shortest paths. (25)

2. A treasure-hunt gaming company is designing a virtual maze where players start at an entry point and try to reach the treasure hidden in one of the chambers. The maze is represented as a grid of cells, and each cell may be either open (free to move) or blocked (a wall). This grid can be modeled as a graph where each open cell is a vertex and edges connect adjacent cells (up, down, left, right). Your task is to implement a Depth-First Search (DFS) based algorithm to explore all possible paths from the entry point to the treasure location.

   The program must determine whether a path exists, and if so, output at least one valid path as a sequence of grid coordinates. Additionally, it should count the total number of distinct paths from entry to treasure that do not revisit the same cell, since players may want to know how many different routes they could possibly take. If no path exists, the program must report accordingly.

For example, in a $4 \times 4$ maze where 0 represents an open cell and 1 represents a wall, with entry at $(0,0)$ and treasure at $(3,3)$, DFS should be used to backtrack through all possible moves, exploring and discarding dead ends, and output results such as: one valid path

$$(0,0) \to (1,0) \to (2,0) \to (3,0) \to (3,1) \to (3,2) \to (3,3),$$

with a total of 3 distinct paths found from entry to treasure. (25)