Week 1 Report:

Network Generation Logic

- **100 students were divided into 4 classes.**
- **Each student is a node in an undirected weighted graph.**
- **Friendship (edges) is more likely:**
  - **Between students of the same class.**
  - **For popular students (using exponential distribution).**
  - **If students share mutual friends (clustering effect).**
- **Friendship strength (edge weight) ranges from 1 (strongest) to 10 (weakest).**

Graph was stored as an adjacency list with weights.

Friend Group Analysis

Using BFS:

- **Number of friend groups (connected components): 4**
- **Smallest group size: 2**
- **Largest group size: 61**

This indicates a semi-fragmented network with one large social core and smaller isolated groups.

Sample Shortest Paths

Using Dijkstra's Algorithm:

Example Pair: 31 → 77
Cost: 11
Path: [31, 45, 77]

Other paths were similarly computed for 5 random pairs.

A* vs Dijkstra (Same pair: 31 → 77)

- **Heuristic: If same class → 0, else → 5**
- **A* cost: 11 | Path: [31, 45, 77]**
- **Dijkstra cost: 11 | Path: [31, 45, 77]**

Result: Both gave the same result, showing A* is effective even with a simple class-based heuristic.

Final Reflection

This project showcased how basic graph algorithms can simulate realistic social structures. By encoding class proximity, popularity, and clustering, we built a model that resembles

actual student dynamics. Shortest paths allowed us to measure closeness, and friend group detection highlighted how connected or isolated different parts of the network are. These tools make the invisible fabric of social life visible and measurable.