**Data Structures and Algorithms**

**Deliverable 1: Project Proposal & Conceptual Design (CLO-3)**

**Group Members: Warda, Abdullah, Asjid**

**1. Project Concept and Problem Description**

**Option 1: Intrusion Maze – Pathfinding + Cyber Defense Simulation**

**Concept:**  
The project simulates a computer network as a maze or graph, where a hacker (attacker) attempts to reach a target server, while the defender traces or blocks routes to prevent intrusion. The simulation aims to visualize network traversal, intrusion detection, and firewall blocking through data structure operations.

**Problem Addressed:**  
Modern cyber-attacks often involve lateral movement across systems. This simulation models how intrusions propagate and how defensive systems can react using algorithmic pathfinding, representing real-time decision-making in cybersecurity.

**Option 2: Mini Cyber Range Game**

**Concept:**  
This is a turn-based cybersecurity strategy game, playable as single or multiplayer. Each player takes the role of either an attacker or defender, managing network nodes (represented as a graph). Attackers exploit or move laterally, while defenders patch vulnerabilities and monitor activities. The goal is to capture or protect high-value targets using limited resources.

**Problem Addressed:**  
The project aims to simulate cyber warfare decision-making in a controlled environment. It gamifies network defense to help visualize how attacks propagate and how layered defenses are structured.

**2. Key Data Structures to be Implemented**

| **Data Structure** | **Purpose in Project** |
| --- | --- |
| **Array / Matrix** | Represent the network grid or list of active systems. |
| **Graph (Adjacency List / Matrix)** | Represent network nodes and their interconnections. |
| **Queue** | Used for breadth-first search (BFS) to simulate hacker movement or action queue. |
| **Stack** | Used for depth-first search (DFS) to simulate defender path tracing, rollback, or undo history. |
| **Hash Table** | Store visited nodes, credentials, vulnerabilities, or player statistics. |
| **Tree** | Represent hierarchical systems (e.g., server → subsystem → endpoint). |
| **Linked List** | Store logs dynamically, such as packet transmission or player actions. |

*(Data structures may be adjusted or optimized during the implementation phase.)*

**3. Main Algorithms**

| **Algorithm** | **Usage / Role in Project** |
| --- | --- |
| **Breadth-First Search (BFS)** | Simulate shortest attack paths and scanning of reachable nodes. |
| **Depth-First Search (DFS)** | Trace defensive routes, rollback detection, or deep traversal. |
| **Hashing Algorithms** | Efficient lookup for visited systems or stored credentials. |
| **Graph Traversal / Pathfinding (Dijkstra or A\*)** | (Advanced phase) Model optimized attacker routes or defensive blocking strategies. |
| **Sorting Algorithms (optional)** | Organize logs or prioritize system alerts. |

**4. Data Flow: Input, Processing, and Output**

**Input:**

* User commands (e.g., move, defend, patch, exploit).
* Network configuration (number of nodes, connections, vulnerabilities).

**Processing:**

* Apply algorithms (BFS/DFS/Dijkstra) to simulate movements or attacks.
* Use data structures to maintain state (graph for network, queue/stack for actions).
* Update network status dynamically after each move.

**Output:**

* Visual or text-based representation of the network grid.
* Logs showing movements, blocked paths, or successful intrusions.
* Game or simulation summary showing who won (attacker/defender).

**5. Integration of Data Structures and Algorithms (DSA Concepts)**

This project integrates **70–80% of the course concepts** by combining theoretical data structures with applied cybersecurity logic:

* **Graph Theory** for representing complex network connections.
* **Stacks and Queues** for traversals, undo/redo actions, and sequential event simulation.
* **Hash Tables** for constant-time data retrieval of nodes or system properties.
* **Linked Lists** for dynamic and efficient logging systems.
* **Trees** for hierarchical data representation.
* **Algorithmic Thinking** (BFS, DFS, Dijkstra) for simulating realistic cyber scenarios.

The project thus demonstrates how data structure design directly impacts performance, scalability, and efficiency in cybersecurity modeling.

**Expected Outcome**

A functional prototype that allows visual simulation of attack-defense scenarios in a networked environment, showcasing core DSA principles through interactive cybersecurity dynamics.