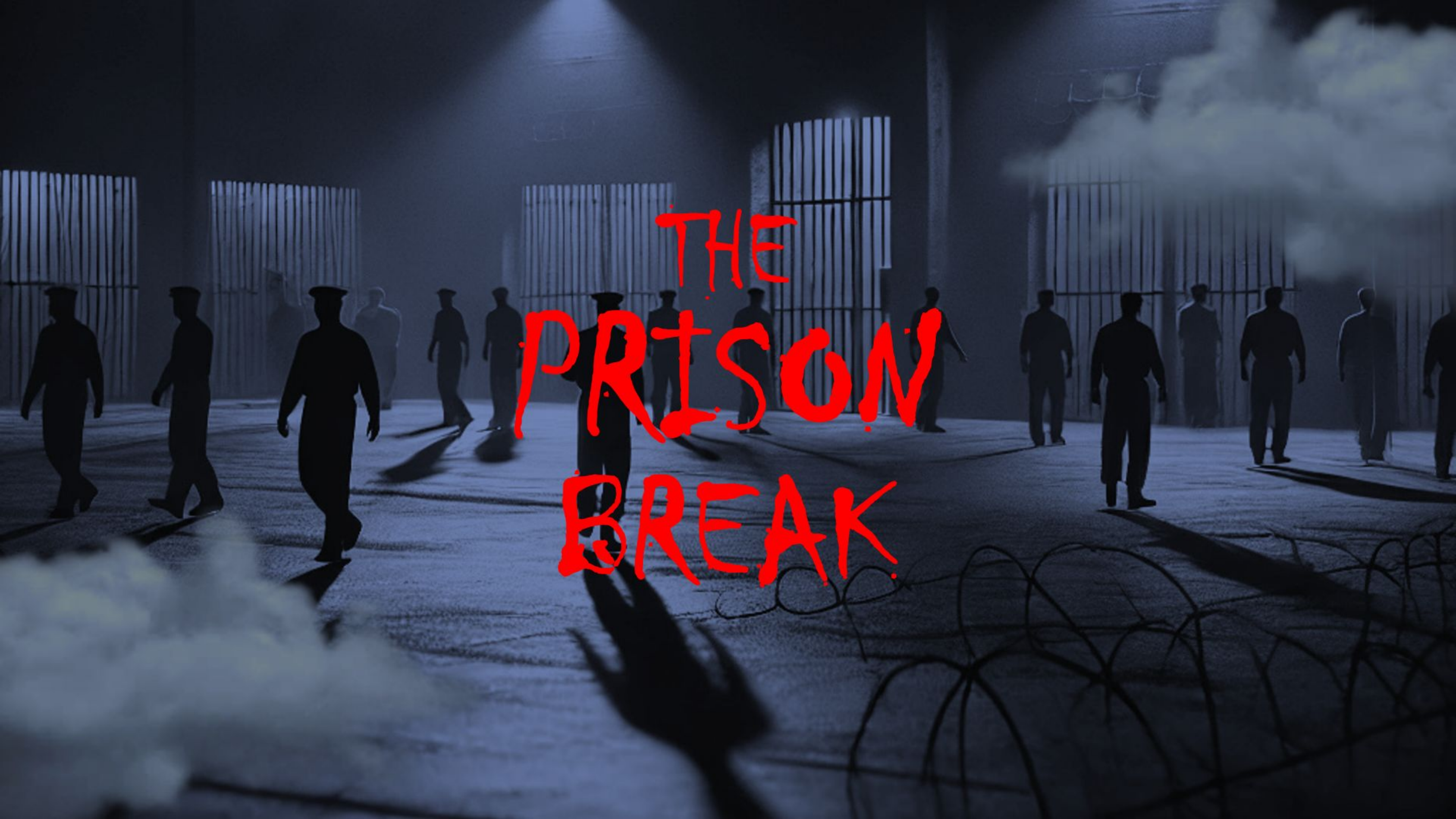
Discrete Structures

Instructor: Mudassir Shabbir

Show and Tell

Presenting by

- Toheed Ali
- Hussnain Javaid
- Taha Qureshi
- M. Ibraheem Tahir
- Bilawal Nouman


THE
PRISON
BREAK

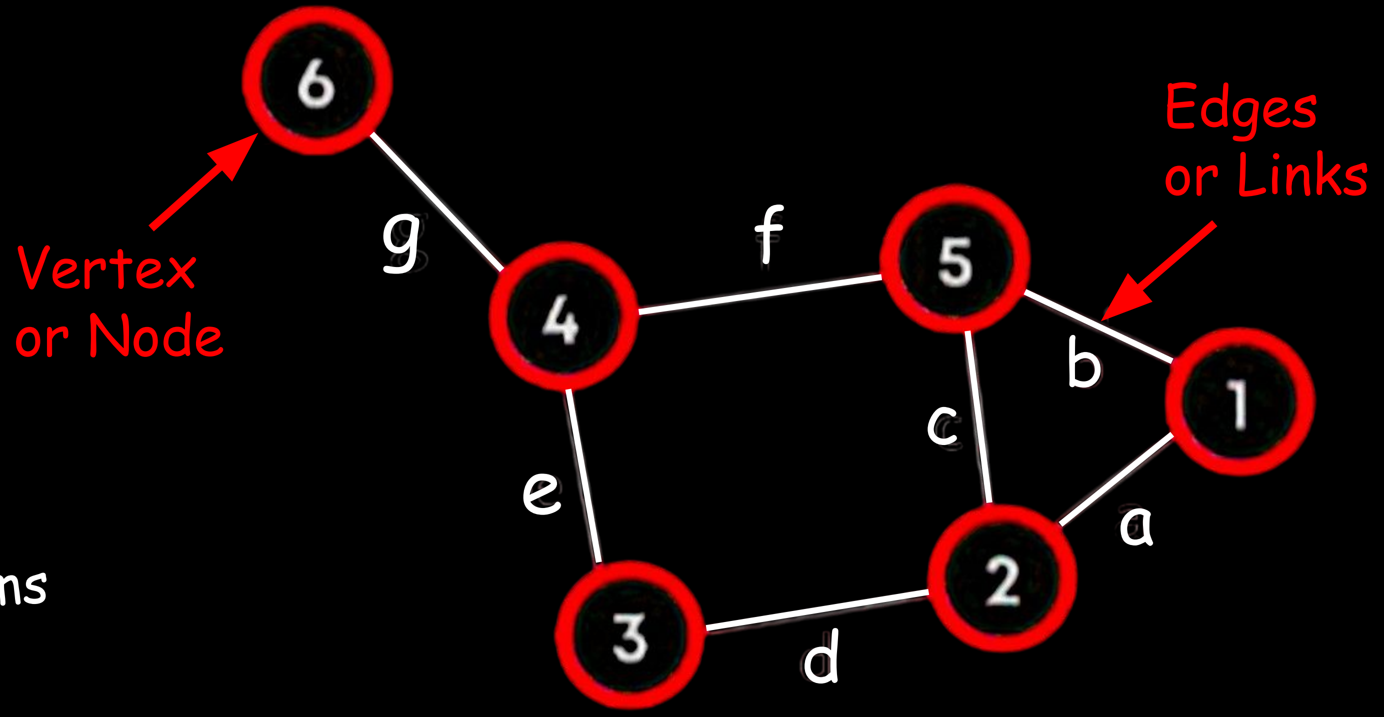
Agenda

1. Introduction to Graph Theory
2. Prison Break Concept in Graph Theory
3. Set theory
4. Prison Break through Logic and Propositions
5. Combination and Permutations
6. Path Finding Algorithm (BFS and DFS)
7. Comparison of Algorithm
8. Live demo of Prison Break (Python Code)
9. Real_World Application
10. Thought Provoking Questions & Discussion

Introduction to Graph Theory

Graph Theory

Graph theory is a branch of discrete mathematics. It is the study of relationships or connections between things which are made up of points connected by lines



Key Components

- **Vertices (Nodes):** Points or objects that are connected; in "Prison Break," they represent rooms or key points. e.g., { 1, 2, 3, 4, 5, 6 }
- **Edges (Links):** Connections between vertices; they represent pathways or links between rooms in the prison layout. e.g., { a, b, c, d, e, f, g }

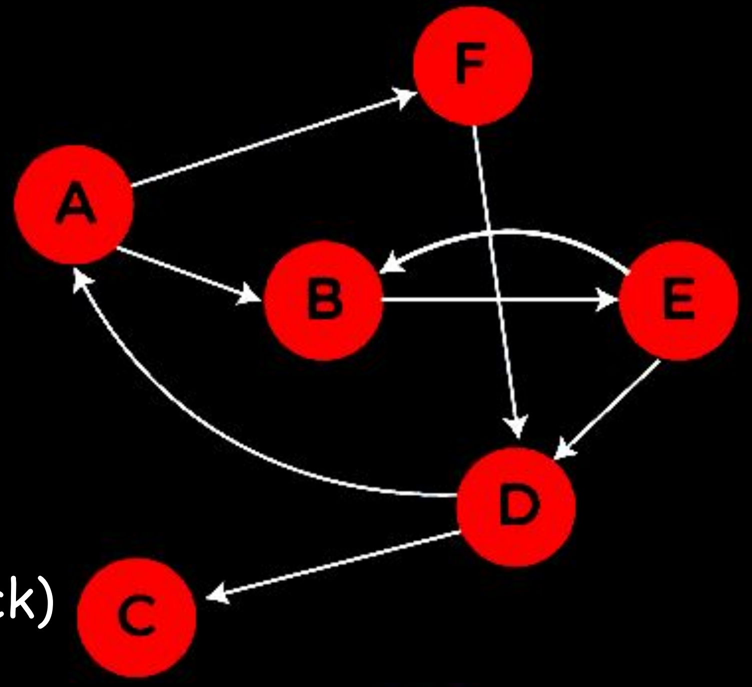
Types of Graph

1 Directed Graph

A directed graph consists of nodes connected by edges, where each edge has a specific direction (from one node to another).

- "Cell A" → "Common room" (one-way path)
- "Common room" → "Exit" (one-way path)

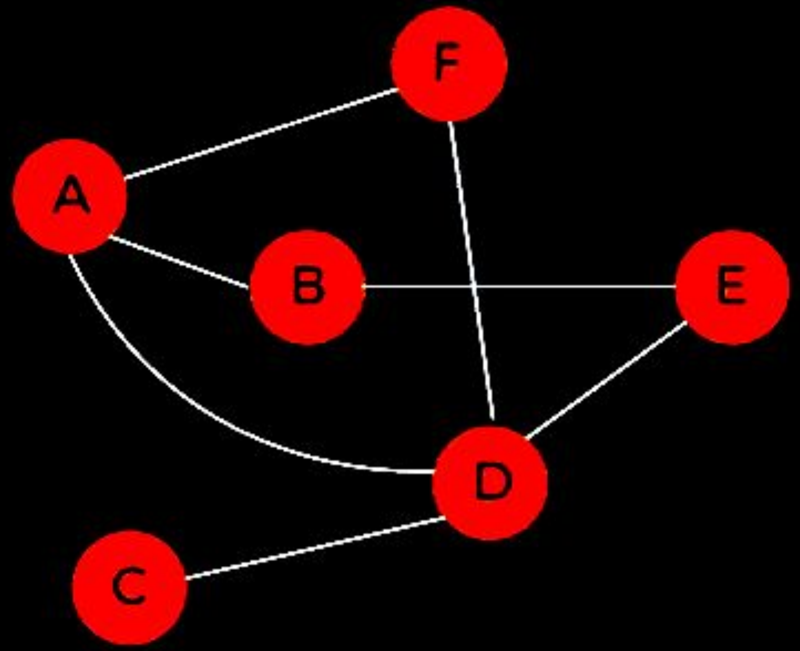
  But no edge from "Exit" → "Common room" (because you can't go back)



2 Undirected Graph

Undirected graph consists of nodes connected by edges that have no direction, meaning you can move freely in either direction between two connected nodes.

- "Cell 1" ↔ "Cell 2" (bidirectional path)
- "Common room" ↔ "Cell 3" (bidirectional path)



Set Theory in Prison Break

Sets and Elements:

- **Location:**

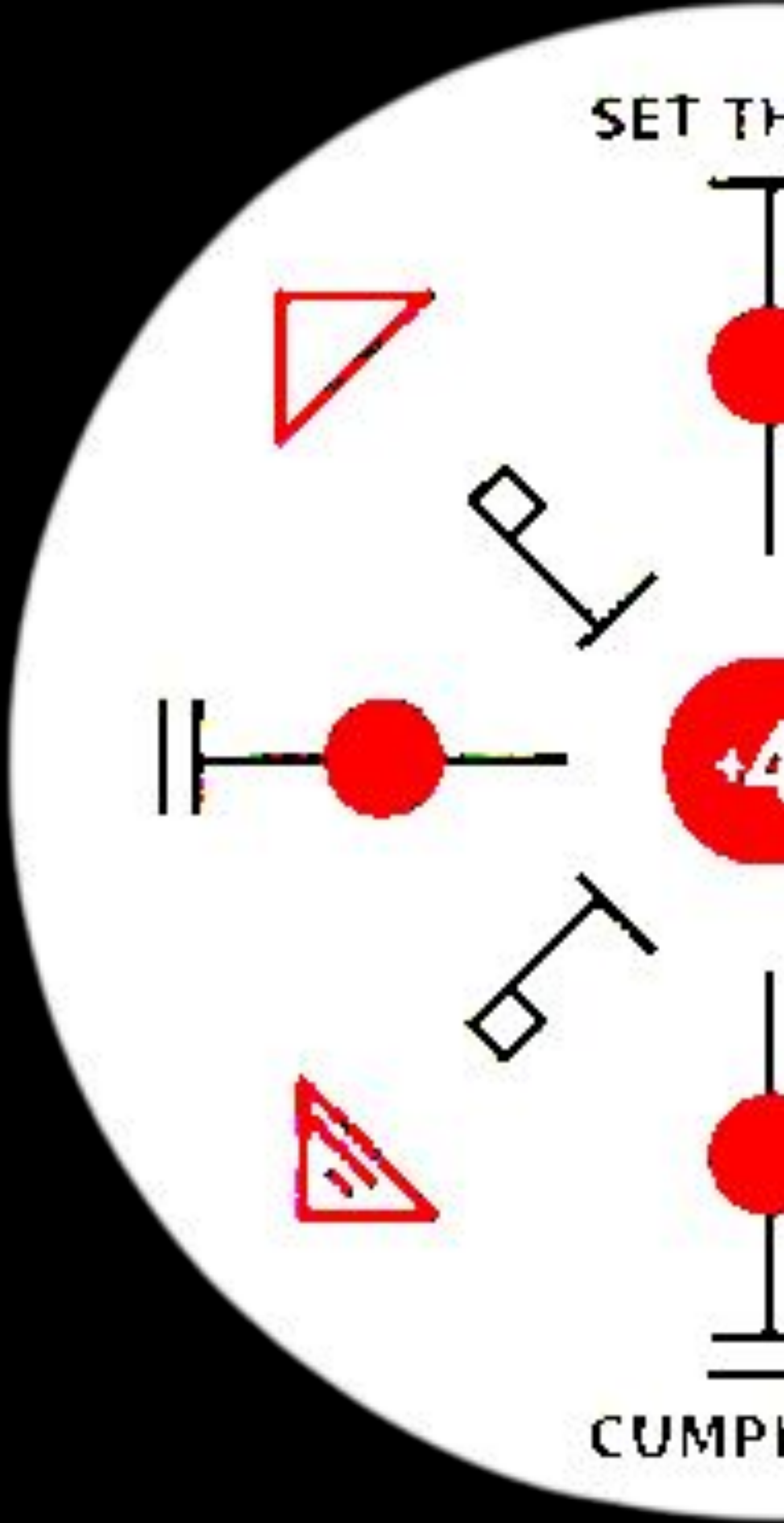
Let L be the set of all locations in the prison.

L = { Cell 1, Cell 2, Cell 3, Common Room, Enter, Exit, Watch Tower}

- **Prisoners:**

Let P = { p_1, p_2, p_3 } where:

  - p_1 represents **Prisoner A** (in Cell 1).
  - p_2 represents **Prisoner B** (in Cell 2).
  - p_3 represents **Prisoner C** (in Cell 3).



Set Theory in Prison Break

Sets and Elements:

- **Guards:**

Let $G = \{ g_1, g_2, g_3, g_4 \}$ where:

g_1 monitors between **Cell 2** → **Enter** repeatedly.

g_2 monitors between **Cell 3** → **Exit** repeatedly.

g_3 moves between **Guard 1's Location** → **Guard 2's Location** → **Common Room.**

g_4 alternates between **Enter** → **Exit** repeatedly from the **Watchtower**

Set Theory in Prison Break

Sets and Elements:

- **Edges (Connections):**

  Let $E \subseteq L \times L$ be the set of edges representing valid paths between locations:

  E = { (Cell 1, Common Room), (Cell 2, Common Room),
  (Cell 3, Common Room), (Common Room, Enter),
  (Common Room, Exit), (Common Room, Watch Tower) }

- **Time:**

  Let T represent the set of all possible times:

  T = { Set of all possible times }
  Cell gates open at Lunch time or Play time

Logics and Proposition

Rule 1: Movement Constraint

A prisoner P_i can move from location x to location y at time t if:

1. There is a direct edge between x and y:

$$E(x, y)$$

2. y is not monitored by any guard at that time:

$$\neg \exists k \, G_k(y, t)$$

Formal Representation

$$\forall t \in T, \forall P_i \in P, \forall x, y \in L, (P_i(x, t) \wedge E(x, y) \wedge \neg \exists k \, G_k(y, t)) \rightarrow P_i(y, t+1)$$

Logics and Proposition in Prison Break

Rule 2: Cell Gate Access

A prisoner can leave their cell Cell_i only if the cell gate is open at time t

Formal Representation

$$\forall t \in T, \forall P_i \in P, (P_i(\text{Cell}_i, t) \land O_{\text{cell}_i}(t)) \rightarrow P_i(\text{Common Room}, t+1)$$

Where:

- $\text{Cell}_i \in \text{Cell}$
- $\text{Cell} = \{\text{Cell}_1, \text{Cell}_2, \text{Cell}_3\}$
- $O_{\text{cell}_i}(t)$ means that the cell is open at a particular time t.

Logics and Proposition in Prison Break

Rule 3: Escape Conditions

Prisoner P_i escapes successfully if they reach Exit at time t without being detected

Formal Representation

$$\forall P_i \in P, \exists t \in T \left(\text{Escape}_i(\text{Exit}, t) \wedge \neg \exists k \, G_k(\text{Exit}, t) \right)$$

Where:

Escape$_i$ is a logical predicate that expresses whether prisoner P_i is present at the designated Exit location at a specific time t

Logics and Proposition in Prison Break

Movement to the Common Room:

- Prisoner P_1 starts at Cell1, and the gate $O_{cell1}(t_0)$ opens at time t_0. Since

  $\neg G_3(\text{Common Room}, t_0)$, P_1 can move to the Common Room:

  $$P_1(\text{Cell1}, t_0) \land O_{cell1}(t_0) \land \neg G_3(\text{Common Room}, t_0) \implies P_1(\text{Common Room}, t_0 + 1).$$

- Similarly, P_2 and P_3 can move. That is:

  $$P_i(\text{Cell}, t_0) \land O_{celli}(t_0) \land \neg G_3(\text{Common Room}, t_0) \implies P_i(\text{Common Room}, t_0 + 1).$$

Where i represents the respective prisoner ($i = 2, 3$).

Logics and Proposition in Prison Break

Moving to the Exit:

- **From the common room, prisoners can move to the Exit if:**

    1. The Exit is not being monitored by G_2 or G_3
    2. E (Common Room,Exit)

- For P_1

$$P_1(\text{Common Room}, t_1) \land \neg(G_2(\text{Exit}, t_1) \land G_4(\text{Exit}, t_1)) \implies P_1(\text{Exit}, t_1 + 1)$$

- Similarly For P_2 and P_3

$$P_i(\text{Common Room}, t_1) \land \neg(G_2(\text{Exit}, t_1) \lor G_4(\text{Exit}, t_1)) \implies P_i(\text{Exit}, t_1 + 1)$$

where $i = 2, 3$.

Logics and Proposition in Prison Break

Combined Equation:

$$P_i(\text{Cell}, t_0) \land O_{celli}(t_0) \land \neg G_3(\text{Common Room}, t_0) \implies$$

$$P_i(\text{Common Room}, t_1) \land \neg(G_2(\text{Exit}, t_1) \lor G_4(\text{Exit}, t_1)) \implies$$

$$P_i(\text{Exit}, t_1 + 1)$$

where i = 1, 2, 3

Logics and Proposition in Prison Break

Proving the Escape:

All prisoners escape successfully if:

$$\exists t, \left(\bigwedge_{i=1}^{3} \left(O_{cell_i}(t) \land \neg G_k(\mathrm{Exit}, t) \right) \land \bigwedge_{i=1}^{3} P_i(\mathrm{Exit}, t) \right)$$

Combination and Permutation

Introduction

Combinations and permutations help calculate different ways to arrange or select escape routes and movements, considering order and grouping. In "Prison Break," they enable us to examine possible sequences of moves and assess success probabilities. By calculating permutations, we simulate various escape routes. This approach provides a structured way to explore all potential actions for effective escape planning.

Combination and Permutation

Arrangement of Prisoners

We have 3 prisoners, each of whom can escape alone, with another, or all together, as the prisons are connected in the tentative map design.

- There are 7 ways the prisoners can escape in combinations we use:

$$^3C_1 +^3 C_2 +^3 C_3 = 3 + 3 + 1 = 7$$

- There is also a possibility that no one escapes we will add it:

$$^3C_1 +^3 C_2 +^3 C_3 +^3 C_0 = 3 + 3 + 1 + 1 = 8$$

Hence we have 8 combinations.

Combination and Permutation

Configuration of guards movement:

As the movement of guards have been already defined:
We know that each guard can move/watch to 2 separate areas (rooms)

Each of these has 2 possible states (either at one position or another).

Hence total configurations of guards movements are :

$$2 \times 2 \times 2 \times 2 = 16$$

Combination and Permutation

Total Outcomes

We know that if we don't stop the prisoners' movement limit, they will move in an infinite loop. To counter this, I have used an 8-step limit for the prisoners.

According to my observations

$$\text{Outcomes} = \text{Guards movement} \times \text{arrangements of prisoners} \times \text{total movement possibility}$$

$$\text{Outcomes} = 16 \times 8 \times 8 = 1024$$

Probability

Possible Escape Outcomes

In our **Prison Break** project, probability is used to calculate the risk factor during escape and estimate success rate. Continuing from the combinatorics calculation, we have 1024 possible paths. To estimate the probability of the prisoners escaping, we assume that, with 3 guards and a watchtower, the prisoners succeed in escaping about 3 out of 100 times.

So, we calculate the probability as:

$$\frac{1024 \times 3}{100} = 30.72$$

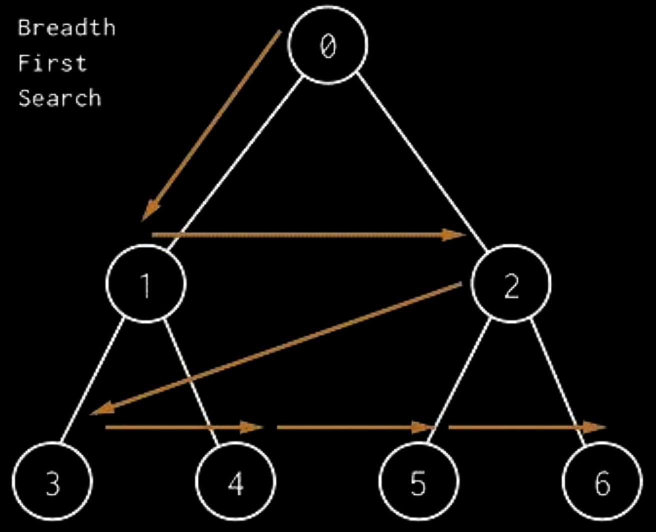
Hence, in approximately 31 out of 1024 attempts, the prisoners would succeed, while in about 993 attempts, they would be caught or unable to escape.

Path Finding Algorithms

Breadth-First Search (BFS)

BFS is a graph traversal algorithm that explores all neighbor nodes at the present depth before moving to the next depth level.

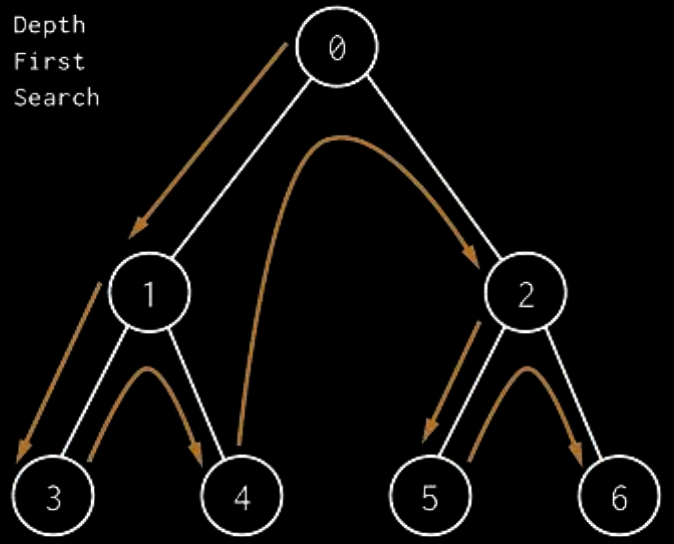
In a prison break scenario, it can be used to find the shortest path to the exit by systematically exploring all possible routes.



Depth-First Search (DFS)

Depth-First Search (DFS) is a graph traversal algorithm that explores as deep as possible along a branch before backtracking.

In prison break scenario it could be used to explore different escape routes, going deep into a particular path before trying another.



Comparison of Algorithms

Time Complexity	BFS: O(V+E), DFS: O(V+E)
Space Complexity	BFS: O(V), DFS: O(V)
Completeness	BFS: Guaranteed to find the shortest path DFS: Not guaranteed
Memory Usage	BFS: Requires more memory DFS: Requires less memory

Live demo of Prison Break (Code)

1

Finding the shortest escape route from the start (S) to the exit (E) while avoiding walls (#) and guards (G) using **Breadth-First Search (BFS)**.

2

- **S:** Start (Prisoner's initial position)
- **E:** Exit (Escape point)
- **#:** Wall (Impassable area)
- **G:** Guard (Guarded area)
- .: Free Space (Path)

3

Prison Grid: The layout of the prison is represented as a 2D list, with each cell containing a character representing walls, guards, or free space.

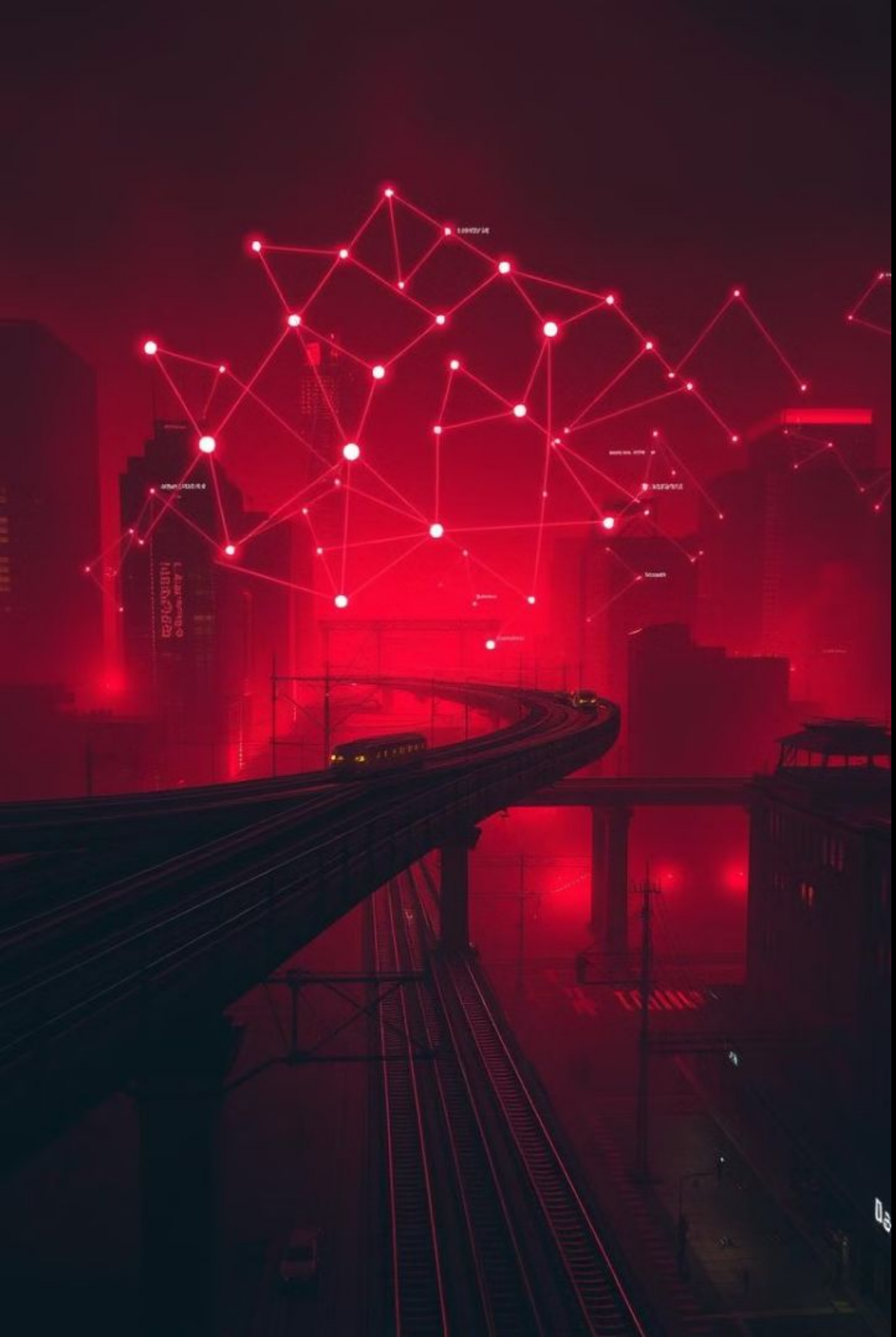
Real-World Application

1 Navigation and Routing Systems

- **Example**: Google Maps, Uber.
- **Use**: Graphs represent locations (nodes) and roads (edges) for optimal routing.
- **Impact**: Saves time and reduces fuel by providing the fastest routes.

2 Computer Networks

- **Example**: Internet traffic.
- **Use**: Algorithms find the fastest paths for data packets.
- **Impact**: Ensures quick and reliable data transmission.



Real-World Application

3 Social Network Analysis

- **Example**: Facebook, LinkedIn.
- **Use**: Graphs connect users (nodes) and relationships (edges) for friend recommendations.
- **Impact**: Enhances user engagement and community building.

4 Traffic Signal Control Systems

- Example: Smart city traffic management.
- Use: Optimizes traffic light timings using graph-based algorithms.
- Impact: Reduces congestion and improves traffic flow.

Conclusion:

By using ideas from discrete mathematics, inmates can plan a prison break in an organized way. This presentation showed how graph theory, set theory, logics and combinations and permutation can help find and use weak points in the prison system.

thank you!