**DISCRERE MATHEMATICS PROJECT**



**Session 2023 - 2027**

**Submitted by:**

Ahmad Bin Rashid 2023-CS-53

Hamza Farooq 2023-CS-72

**Assigned by:**

Sir Waqas Ali

**Course:**

CSC-101 Discrete Mathematics

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

Table of Contents

[Individual Contribution: 4](#_Toc155200768)

[Analysis of Games using Graph Theory: 4](#_Toc155200769)

[Analysis of Tic-Tac-Toe: 4](#_Toc155200770)

[Graph Representation: 4](#_Toc155200771)

[ Nodes (Vertices): 4](#_Toc155200772)

[ Edges: 4](#_Toc155200773)

[What We Are Trying to Achieve: 4](#_Toc155200774)

[ Winning States: 4](#_Toc155200775)

[ Avoiding Losing States: 4](#_Toc155200776)

[Generalizations and Complexity: 5](#_Toc155200777)

[Minimax Algorithm: 5](#_Toc155200778)

[Avoiding Cycles: 5](#_Toc155200779)

[Analysis of Dots and Boxes: 5](#_Toc155200780)

[Graph Representation: 5](#_Toc155200781)

[ Nodes (Vertices): 5](#_Toc155200782)

[ Edges: 5](#_Toc155200783)

[What We Are Trying to Achieve: 6](#_Toc155200784)

[ Completing Boxes: 6](#_Toc155200785)

[ Avoiding Opponent's Moves: 6](#_Toc155200786)

[Strategy and Analysis: 6](#_Toc155200787)

[ Graph Structure: 6](#_Toc155200788)

[ Node Degrees: 6](#_Toc155200789)

[ Game Complexity: 6](#_Toc155200790)

[Graph Algorithms: 6](#_Toc155200791)

[ Finding Winning Moves: 6](#_Toc155200792)

[ Avoiding Losing Moves: 6](#_Toc155200793)

[Generalizations: 6](#_Toc155200794)

[Analysis of Chess 7](#_Toc155200795)

[Graph Representation: 7](#_Toc155200796)

[ Nodes (Vertices): 7](#_Toc155200797)

[ Edges: 7](#_Toc155200798)

[What We Are Trying to Achieve: 7](#_Toc155200799)

[ Reachability 7](#_Toc155200800)

[ Optimal Paths: 7](#_Toc155200801)

[ Game Complexity: 7](#_Toc155200802)

[Analyzing Chess Positions: 7](#_Toc155200803)

[ Check and Checkmate: 7](#_Toc155200804)

[ Game Trees: 8](#_Toc155200805)

[Minimax Algorithm: 8](#_Toc155200806)

[Avoiding Cycles: 8](#_Toc155200807)

[Generalizations: 8](#_Toc155200808)

Individual Contribution:

This project of Discrete Mathematics is made by the collective efforts of **Ahmad Bin Rashid** and **Hamza Farooq**. In this project Ahmad Bin Rashid has contributed in solving graph theory programs, analysis of tic tac toe and the slides for video demonstration. Whereas, Hamza Farooq has written programs for Number Theory Questions. And some work on slides is done by him. For all the questions, concepts were discussed. Overall, the project was completed by the collective efforts of both members.

Analysis of Games using Graph Theory:

* **Analysis of Tic-Tac-Toe**
* **Analysis of Dots and Boxes**
* **Analysis of Chess**

Analysis of Tic-Tac-Toe:

Tic-Tac-Toe can indeed be represented as a game graph. The graph for Tic-Tac-Toe is a state space graph, where each node represents a unique game state, and the edges between nodes represent legal moves. The goal is to achieve a winning state or, in the case of Tic-Tac-Toe, to achieve three in a row horizontally, vertically, or diagonally.

# Graph Representation:

* Nodes (Vertices): Each node in the graph represents a unique configuration of the Tic-Tac-Toe board. The configuration includes the positions of Xs and Os.
* Edges: Directed edges represent legal moves from one state to another. Each edge corresponds to placing an X or an O in an empty cell on the board.

# What We Are Trying to Achieve:

* Winning States: The goal is to reach a winning state, where one player has three in a row horizontally, vertically, or diagonally. In the graph, these winning states are terminal nodes.
* Avoiding Losing States: Players aim to avoid reaching states where their opponent has three in a row.

# Generalizations and Complexity:

Tic-Tac-Toe can be generalized to larger board sizes or different win conditions. For example, a generalized version might be played on an n×n board where a player wins by having k in a row. The game graph for this generalized version becomes more complex as the branching factor increases with the larger board size.

# Minimax Algorithm:

Graph theory is also used to analyze and optimize strategies for Tic-Tac-Toe. The Minimax algorithm, which is widely used in game theory, searches through the game graph to determine the best move for a player. It involves evaluating each possible move by recursively exploring the game tree to a certain depth and assigning a value to each terminal node (win, lose, or draw). The algorithm then selects the move with the highest value if it's the maximizing player's turn and the lowest value if it's the minimizing player's turn.

# Avoiding Cycles:

The game graph for Tic-Tac-Toe is acyclic since a repeated board configuration doesn't make sense in the context of the game. A cycle would imply a repeated state, which is not allowed.

In summary, representing Tic-Tac-Toe as a graph provides a structured way to analyze the game's state space, determine winning or losing positions, and develop optimal strategies using algorithms like Minimax. It also allows for generalizations and variations of the game by adjusting the size of the board or win conditions

**Analysis of Dots and Boxes:**

Dots and Boxes (also known as the Dot Game or Squares) is a pencil-and-paper game for two players. The game starts with an empty grid of dots. Players take turns connecting two horizontally or vertically adjacent dots, one line at a time. The player who completes the fourth side of a 1x1 box earns one point and takes another turn. The game ends when no more lines can be added. The player with the most completed boxes at the end wins.

# Graph Representation:

* Nodes (Vertices): Each node represents a dot on the grid.
* Edges: Edges connect horizontally or vertically adjacent dots. Each edge represents a potential line that a player can draw.

# What We Are Trying to Achieve:

* Completing Boxes: The goal is to complete as many boxes as possible. Completing a box gives the player a point and an additional turn.
* Avoiding Opponent's Moves: Players strategize to prevent their opponents from completing boxes and gaining points.

# Strategy and Analysis:

* Graph Structure: The graph is a grid of dots where edges connect adjacent dots. The size of the grid determines the complexity of the game.
* Node Degrees: The degree of each node (dot) represents the number of lines connected to it. In a standard game, nodes typically have a degree of 2.
* Game Complexity: The size of the grid affects the branching factor of the game graph. Larger grids result in a more complex state space.

# Graph Algorithms:

* Finding Winning Moves: Algorithms can be applied to find winning moves or optimal strategies. For instance, a player may try to create loops to force the opponent to complete a box.
* Avoiding Losing Moves: By analyzing the graph, players can identify moves that lead to certain loss or disadvantage.

# Generalizations:

Dots and Boxes can be generalized to larger grids or modified rules, leading to variations of the game. A larger grid increases the complexity of the game graph and provides more strategic possibilities.

In summary, representing Dots and Boxes as a graph allows for the analysis of game states, identification of winning and losing positions, and the development of optimal strategies. The graph structure captures the connectivity of dots and lines, providing insights into the dynamics of the game.

Analysis of Chess

Chess can be represented and analyzed using graph theory, specifically by modeling the game board as a graph. In this representation, each square on the chessboard corresponds to a vertex, and the legal moves of a piece define the edges between vertices. Here's how chess can be approached from a graph theory perspective:

# Graph Representation:

* Nodes (Vertices): Each square on the chessboard is a node in the graph. For a standard 8x8 chessboard, there are 64 nodes.
* Edges: Edges between nodes represent legal moves of chess pieces. Different pieces have different move patterns, and the edges capture these movements. For example:
* A knight moves in an "L" shape, resulting in edges connecting a knight's starting square to its potential landing squares.
* Pawns, bishops, rooks, queens, and kings have their own distinct move patterns, and the edges reflect those patterns.

# What We Are Trying to Achieve:

* Reachability: Determine if a square can be reached from another square by legal moves of a given piece.
* Optimal Paths: Find the shortest path for a piece to reach a particular destination.
* Game Complexity: Analyze the complexity of the game by considering the number of possible moves at each position.

# Analyzing Chess Positions:

* Check and Checkmate: Graph theory can be used to identify whether a king is in check or checkmate. If there is a path from an opponent's piece to the king's square, the king is in check.
* Game Trees: Chess games can be represented as game trees, where each node represents a game position, and edges represent legal moves. This facilitates the analysis of possible moves and outcomes.

# Minimax Algorithm:

The Minimax algorithm, often used in game theory, can be applied to analyze chess positions. Minimax evaluates possible moves by exploring the game tree to a certain depth, assigning values to terminal nodes (win, lose, or draw), and selecting the move that maximizes the value for the player and minimizes it for the opponent.

# Avoiding Cycles:

The game graph for chess is acyclic, as a repeated board configuration during a game doesn't make sense in the context of chess. A cycle would imply a repeated state, which is not allowed.

# Generalizations:

Chess can be generalized to different board sizes or variations of the rules. The graph representation remains applicable to these variations, allowing for a flexible analysis of different forms of the game.

In summary, representing chess as a graph provides a formal and structured way to model the relationships and connections between different squares on the board. This allows for the analysis of reachability, optimal paths, and overall game complexity, as well as the application of algorithms like Minimax for strategic decision-making