EF234405 Design & Analysis of Algorithms (D)
Quiz 2

No.
Date

"In the name of Allah (God) Almighty, I hereby pledge and sincerely declare that I have completed Quiz 2 independently. I have not committed any form of cheating, plagiarism, or received unauthorized assistance. I accept all consequences should it be proven that I have engaged in cheating and/or plagiarism."

Surabaya, 29 May 2025

<Signed>

[Ammar Ghozy Tanumijaya]
[5025231203]

<Signed>

[Huraira Shehmue Mahaneel]
[5025231216]

Percentage of Contribution
and
Specific Roles

- Ghozy : ~~Perform analysis, testing, and~~ wrote the final report.
- ~~Shen : Design the user interface, developed the main structure, and implemented~~ the algorithm.

1. **Form a Group**
    - Ammar Ghozy Tanumijaya 5025231203
    - Huraira Shenmue Mahanee 5025231216

2. **Develop a computer program**

We developed a browser-based Memory Matching Game that requires players to find 10 pairs of hidden numbers. The game is powered by the Breadth-First Search (BFS) algorithm, which dynamically suggests the optimal move to help the player complete the game efficiently. Features like hints, card previewing, and real-time statistics are designed to enhance both playability and educational value.

3. **Programming Language**

We use :

- **HTML** – to structure the user interface
- **CSS** – to create a clean, responsive design
- **JavaScript** – to manage game logic, animations, user interaction, and BFS implementation

4. **Link to the Github**

Github Project

5. **The Project**

    - *The Project Design*

The game consists of a single-page application featuring:

- A **dynamic game board** rendered using JavaScript with 20 clickable cards (10 matching pairs).
- A **header** with real-time statistics: number of pairs found, elapsed time, and total moves.
- **Interactive controls** including:
    - 💡 **BFS Hint**: Highlights the next optimal matching pair.
    - 👁️ **Peek All**: Temporarily reveals all unmatched cards.
    - 🔄 **New Game**: Resets the board and stats.
- A dedicated **information panel** explaining the algorithm and user actions.

The layout is **responsive** with a mobile-friendly design and interactive visual feedback through animations and color coding.

- *Implementation*

- **Shuffle Logic**

We used the **Fisher-Yates Shuffle** algorithm to randomly distribute card pairs, ensuring fairness and unpredictability.

- **Gameplay Mechanics**

1. Clicking a card reveals its value.
2. Two flipped cards are checked for a match.
3. Matching pairs are locked in place; unmatched ones are flipped back.
4. Game ends when all pairs are matched.

- **BFS Algorithm for Hinting**

The BFS algorithm is central to the “Hint” feature:

- It models the game as a **state space**, where each node represents a set of matched cards.
- It traverses this space level-by-level to compute the **shortest sequence of matching steps**.
- The next optimal match is extracted from this path and visually suggested to the user.

- **Additional Features**

- **Peek All**: A timed function reveals all unmatched cards for memory aid.
- **Game stats**: Tracks moves, time, and dynamically updates the status (“Playing” or “You Won!”).
- **Victory message**: Summarizes performance upon completion.

- *Styling & Interface*

The game’s design prioritizes clarity, usability, and engagement:

- **Color-Coding**:
  - Unflipped cards: neutral
  - Flipped: blue-themed
  - Matched: green
  - BFS Hint: yellow with pulse animation
- **Typography**: Uses modern, readable fonts (`Segoe UI`, `Verdana`)
- **Responsive Design**:
  - Cards resize and reposition for smaller screens
  - Buttons and stats adjust layout for accessibility
- **Visual Feedback**:|
  Transitions, hover effects, and animations improve interactivity

Overall, the CSS ensures a smooth, pleasant experience across devices and screen sizes.

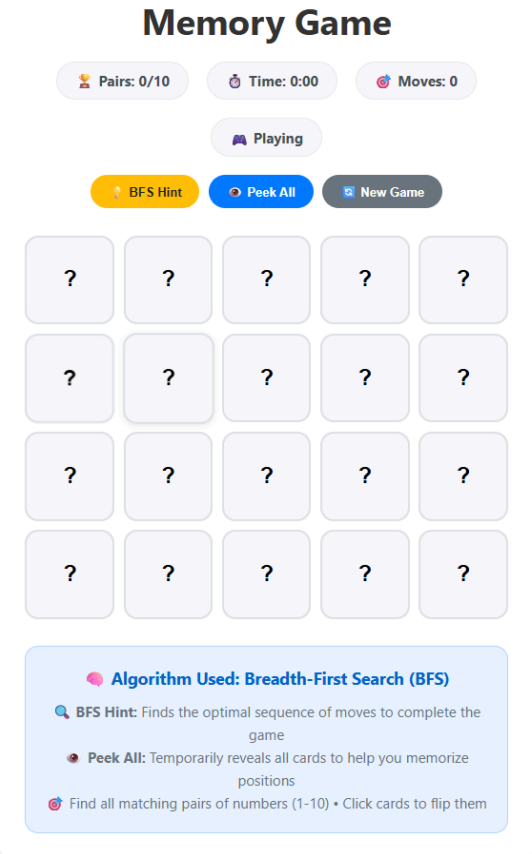
- *Analysis / Evaluation*

- **Strengths:**
  - The BFS algorithm improves gameplay by providing intelligent hints, demonstrating a practical use of AI-style logic in a game.
  - Clear separation of logic (JS), structure (HTML), and style (CSS) shows good software engineering practice.
  - Responsive and intuitive UI enhances user accessibility.

- **Algorithm Efficiency:**
  - BFS is performant for the 20-card configuration, running in real time with no delays.
  - The use of set-based state hashing avoids redundant searches and ensures optimal pathfinding.

- **Learning Outcome:**
  - This project helped us internalize algorithmic thinking and user-centric design.
  - It also provided experience in handling game state transitions, animations, and search-based logic.

  - *The Application*



The screenshot shows the main interface of the **Memory Game**, designed to be clean, responsive, and user-friendly. Here's a breakdown of the layout and elements:

🧠 Header and Game Status

- **Title**: Displayed prominently at the top — *"Memory Game"* — using bold and modern typography.
- **Statistics Bar**: Shows real-time game metrics:
  - 🏆 **Pairs**: Indicates the number of matched pairs out of 10.
  - ⏱️ **Time**: Tracks the elapsed time in minutes and seconds.
  - 🎯 **Moves**: Counts the number of moves made by the player.
- **Game Status Indicator**: Shows a tag like *"🎮 Playing"* or *"🎉 You Won!"* to reflect current progress.

🎮 Control Buttons

- 💡 **BFS Hint** (Yellow): Activates the BFS algorithm to suggest the next optimal move.
- 👁️ **Peek All** (Blue): Temporarily reveals all unmatched cards for 2 seconds.
- 🔄 **New Game** (Gray): Resets the board, timer, and stats to start a new session.

Each button uses distinct colors and icons to enhance visual clarity and functionality.

🧩 Game Board

- A **4x5 grid** containing 20 cards, each initially showing a "?" mark.
- Cards are flipped by clicking and reveal a number (1–10). Two cards form a pair if their numbers match.
- Visual effects like hover shadows, color highlights (e.g., blue for flipped, green for matched), and animations improve interaction feedback.

📘 Algorithm Information Panel

- A highlighted section at the bottom explains the algorithm used:
  - 💡 **BFS Hint**: Describes how BFS is used to calculate the optimal matching sequence.
  - 👁️ **Peek All**: Clarifies the functionality of revealing cards.
  - 🧩 General instruction to match all pairs by clicking cards.

The UI is thoughtfully designed to be:

- **Visually appealing** using soft color schemes and rounded elements.
- **Functional** with intuitive controls and immediate feedback.
- **Informative** by explicitly mentioning the applied algorithm and instructions for users.

- *Conclusion*

This Memory Game project successfully bridges **algorithm theory and real-world application**, showing how **Breadth-First Search** can enrich user experiences. Through careful design and modular coding, we built an engaging and educational game that blends interactivity with strategy. This project not only improved our technical skills but also deepened our understanding of how classical algorithms can enhance everyday software.