**AI Snake Game Assignment**

**A\* Star Search:**

**Implemented By: Ayesha Ulfat**

**Code Description:**The key components of the implementation are:  
**heuristic (a, b):** It calculates the distance between the head of the snake and food.

**AstarSolution ():**

This method computes the optimal path from the snake's current position to the food using the A\* algorithm. It utilizes a priority queue to explore nodes efficiently based on their estimated costs.

**Performance and Results:**  
The A\* Search showed great performance. While testing its performance, the snake never died. I tested it till the score reached 600 and the it was successful in finding the optimal path without hitting an obstacle.

**ALGORITHM : GREEDY\_BFS**

**IMPLEMENTED BY: RIDA MALIK**

**CODE Description:**

**Heuristic Function:**

The heuristic method calculates the Manhattan distance between the snake's head and the food position. This heuristic guides the search towards the direction that minimizes the distance to the food.

**Data Structure:**

The algorithm uses a priority queue (Priority Queue) to store and explore the states. States are prioritized based on the heuristic value, ensuring that the algorithm explores the most promising states first.

**Performance:**

The Greedy BFS approach excelled in guiding the snake through the game, demonstrating its reliability by avoiding all the red walls and achieving a high score of 1060. Its efficient path finding process ensured smooth gameplay and successful movement of the snake.

**ALGORITHM : UNIFORM COST SEARCH**

**IMPLEMENTED BY: NEHAAL ASIF**

**CODE Description:**

**Description:**

This code is a smart way for a snake to find food in a game. It works by trying different paths to reach the food with the least effort. It keeps track of where it's been and how much effort each path takes and than choosing with the minimum cost .

**Algorithm:**

It's like a person exploring a maze, always choosing the easiest path forward with the minimum possible cost. The snake looks at all possible moves and picks the one that costs the least, until it finds the food but due to making decision to go to optimal path it takes some time to make decision optimally.

**Performance:**

Though it's working fine, it might be slow because it carefully considers every move. Just like taking your time to solve a puzzle, it ensures the snake finds the best way to reach the food, even if it takes a bit longer.