#### ****Algorithm Performance Overview:****

**Breadth-First Search (BFS)**

* + **Time Taken:** Moderate
  + **Moves Taken:** Shortest path guaranteed
  + **Failure Rate:** Low

BFS ensures the shortest path but explores all possible directions evenly, making it slower than heuristics-based methods.

**Depth-First Search (DFS)**

* + **Time Taken:** Fast (but inconsistent)
  + **Moves Taken:** Can be longer than necessary
  + **Failure Rate:** High

DFS sometimes gets stuck in long, unnecessary paths, leading to higher failure rates.

**Uniform Cost Search (UCS)**

* + **Time Taken:** Slow
  + **Moves Taken:** Shortest path
  + **Failure Rate:** Low

UCS is optimal like BFS but can be slower due to evaluating all possible moves equally.

**Iterative Deepening Search (IDS)**

* + **Time Taken:** Very Slow
  + **Moves Taken:** Efficient (similar to BFS)
  + **Failure Rate:** Low

IDS explores depth-first but with controlled backtracking, making it slower but effective.

**Greedy Best-First Search (GBFS)**

* + **Time Taken:** Very Fast
  + **Moves Taken:** Often short, but not always optimal
  + **Failure Rate:** Moderate

Greedy BFS follows the most promising path but sometimes gets stuck in local optima.

**A Search (A\*)\***

* + **Time Taken:** Fast
  + **Moves Taken:** Optimal (shortest)
  + **Failure Rate:** Very Low

A\* balances cost and heuristic distance, making it the most efficient.

**Random Move Strategy**

* + **Time Taken:** Unpredictable
  + **Moves Taken:** Very High
  + **Failure Rate:** Extremely High

The random strategy moves blindly and rarely reaches the food efficiently.

### ****Conclusion:****

A\* search is the best overall performer, offering the shortest path in minimal time with low failure rates. BFS and UCS also perform well but are slightly slower. Greedy BFS is fast but unreliable, while DFS and random movement perform poorly.

Project by:-

Garima Das 23057022  
Debasis Mishra 23057019  
Dibya Kirti Pattanaik 23057021