# Project Report: Analysis of Search Algorithms

## 1. Introduction

This project solves the classic "River Crossing Puzzle" using an intelligent agent. The goal is to transport a Farmer, Wolf, Sheep, and Cabbage from the left bank to the right bank without violating specific safety constraints (e.g., Wolf cannot be left with Sheep).

## 2. Problem Formulation

- **States:** Represented as a vector (F,W,S,C) where 0 is Left Bank and 1 is Right Bank.
- **Initial State:** (0,0,0,0).
- **Goal State:** (1,1,1,1).
- **Actions:** Move(Farmer), Move(Farmer + Item).
- **Constraints:**
    - Wolf eats Sheep if Farmer is absent.
    - Sheep eats Cabbage if Farmer is absent.

## 3. Implemented Algorithms

We implemented and compared five search strategies:

1. **Breadth-First Search (BFS):**
    a. **Strategy:** Explores the shallowest nodes first.
    b. **Outcome:** Guaranteed to find the shortest path (7 steps).
2. **Depth-First Search (DFS):**
    a. **Strategy:** Explores the deepest nodes first.
    b. **Outcome:** Found a solution, but required cycle detection to avoid infinite loops.
3. **Uniform Cost Search (UCS):**
    a. **Strategy:** Explores based on path cost g(n).
    b. **Outcome:** Identical to BFS in this problem since step cost = 1.
4. **Iterative Deepening Search (IDS):**
    a. **Strategy:** Repeatedly runs DFS with increasing depth limits.
    b. **Outcome:** Combined optimality of BFS with memory efficiency of DFS.

5. *A Search (A-Star):**
    a. **Strategy:** Uses f(n)=g(n)+h(n).
    b. **Heuristic h(n):** The number of items remaining on the starting bank.
    c. **Outcome:** The most efficient algorithm, directing the search towards the goal intelligently.

# 4. Experimental Results & Comparison

The following table summarizes the performance of the algorithms based on our execution:

| Algorithm | Solution Found? | Path Cost (Steps) | Optimal? | Time Complexity |
|---|---|---|---|---|
| **BFS** | Yes | 7 | ☑ Yes | O(b^d) |
| **DFS** | Yes | 7 (Variable) | ❌ No | O(b^m) |
| **UCS** | Yes | 7 | ☑ Yes | O(b^{C^*/\epsilon}) |
| **IDS** | Yes | 7 | ☑ Yes | O(b^d) |
| **A*** | Yes | 7 | ☑ Yes | Depends on Heuristic |

# 5. Conclusion

While all algorithms successfully solved the puzzle, *A Search** and **BFS** proved to be the most reliable for finding the optimal solution. The GUI developed demonstrates these differences visually, allowing users to observe the traversal order of each strategy.