

## □ 8-Puzzle Solver

### Search Algorithms Project – Python

---

#### Introduction

This project demonstrates the application of **Artificial Intelligence search algorithms** to solve the **8-Puzzle Problem**.

The goal is to arrange the numbers from 1 to 8 on a  $3\times 3$  grid with one empty space to reach the goal state using different search methods.

---

#### Project Features

- **State Representation:** Each state is represented as a  $3\times 3$  matrix or a list of 9 elements (numbers + empty space).
  - **Multiple Search Algorithms:** Implementation of 6 search algorithms.
  - **Performance Evaluation:** Measures nodes explored, solution length, and execution time.
  - **Flexible Initial States:** Supports different starting states for testing.
- 

#### Implemented Search Algorithms

##### Uninformed Search Algorithms

###### 1 Breadth-First Search (BFS)

- Explores all states level by level.
- Guarantees the shortest solution but requires high memory.

###### 2 Depth-First Search (DFS)

- Explores one path deeply before backtracking.
- Low memory usage but may not find the shortest path.

###### 3 Iterative Deepening Search (IDS)

- Applies DFS repeatedly with increasing depth limits.
- Combines BFS optimality with DFS memory efficiency.

## 4 Uniform Cost Search (UCS)

- Expands the node with the lowest cumulative cost first.
  - Guarantees an optimal solution.
- 

## ◆ Informed Search Algorithms

### 5 Hill Climbing

- Always moves toward the neighbor closest to the goal (heuristic).
- Very fast but may get stuck in local optima.

### 6 A\* Search Algorithm

- Uses the evaluation function:  $f(n) = g(n) + h(n)$
  - **Heuristics:**
    - Manhattan Distance:  $|x_1 - x_2| + |y_1 - y_2|$
    - Misplaced Tiles: Number of tiles in the wrong position.
  - Guarantees optimal solution while exploring fewer nodes than BFS.
- 

## II Performance Evaluation (Example)

Algorithm	Moves to Goal	Nodes Visited	Execution Time	Status
BFS	20 (Optimal)	1500	200 ms	Solved
DFS	45	800	150 ms	Solved
UCS	20 (Optimal)	1550	210 ms	Solved
A*	20 (Optimal)	200	15 ms	Solved
Hill Climbing	—	50	5 ms	Failed

---

## ❖ Comparative Analysis

### Time Complexity

- BFS:  $O(b^d)$

- DFS:  $O(b^m)$
- UCS:  $O(b^d)$
- IDS:  $O(b^d)$
- A\*:  $O(b^d)$  (depends on heuristic quality)
- Hill Climbing:  $O(b)$

## Optimality & Cross Path

Algorithm	Optimal Path	Cross Path
BFS	Yes	No
DFS	No	No
UCS	Yes	No
IDS	Yes	No
A*	Yes	No
Hill Climbing	No	Yes

Algorithm	Optimal Path	Cross Path
BFS	Yes	No
DFS	No	No
UCS	Yes	No
IDS	Yes	No
A*	Yes	No
Hill Climbing	No	Yes

---

## Conclusion

- Uninformed search explores the full state space, resulting in higher time and memory usage.
- A\* significantly reduces the number of nodes explored using heuristics while guaranteeing an optimal solution.
- For the 8-Puzzle Problem, A\* is the preferred choice, while Hill Climbing is fast but not guaranteed to find a solution.

---

## Project Author

Mahmoud hany emara