8 Puzzle Problem Report

Project Title: 8 Puzzle Problem Solver Using A* Search Algorithm

Submitted by: Devansh Jadon

Date: 11-03-2025

Branch: CSEAI

University Roll no: 2024011003000097

Submited to: Shivansh Prashad

2. Introduction

What is the 8 Puzzle Problem?

The **8-puzzle problem** is a sliding tile puzzle that consists of a **3x3 grid** containing **eight numbered tiles (1 to 8)** and **one blank space (represented as 0)**. The objective of the puzzle is to rearrange the tiles from an **initial state** to a **goal state** by sliding the tiles **up, down, left, or right** into the blank space.

The puzzle is often used to demonstrate the application of **Artificial Intelligence (AI)** and **search algorithms**. In this project, we use the *A Search Algorithm (A-Star)** to solve the puzzle.

Problem Statement

Given:

- An **initial state** (input from the user).
- A **goal state** (input from the user).

The task is to find the **minimum number of moves** to solve the puzzle and output the step-by-step solution.

---

3. Methodology

Approach Used

In this project, we used the *A Search Algorithm** along with the **Manhattan Distance heuristic** to solve the puzzle.

A* Search Algorithm

- The A* (A-Star) algorithm is a widely used search algorithm in AI that finds the shortest path from the initial state to the goal state.
- It uses two components:
    - **g(n):** Cost from the start node to the current node.
    - **h(n):** Heuristic cost (Manhattan Distance) from the current node to the goal node.

The algorithm minimizes the total cost function:
$f(n) = g(n) + h(n)$

-
- The node with the **lowest f(n)** is expanded first, ensuring an optimal solution.

Manhattan Distance Heuristic

The **Manhattan Distance** calculates the sum of vertical and horizontal moves each tile has to make to reach its goal position. The formula is:

$$h(n) = |x1 - x2| + |y1 - y2|$$

where:

- `(x1, y1)` = current position of the tile.
- `(x2, y2)` = target position of the tile.

Input and Output

- The user is asked to input the **initial state** and **goal state** in a 3x3 grid format.
- The program will then calculate the minimum steps to solve the puzzle.
- It will display the step-by-step output until the goal state is reached.

---

4. Code Typed

Here is the complete Python code to solve the **8 Puzzle Problem**:

```
import heapq

def manhattan_distance(state, goal):
    distance = 0
    for i in range(3):
        for j in range(3):
            if state[i][j] != 0:
                x, y = divmod(goal.index(state[i][j]), 3)
                distance += abs(x - i) + abs(y - j)
    return distance

class Node:
    def __init__(self, state, parent, move, cost, heuristic):
        self.state = state
        self.parent = parent
        self.move = move
        self.cost = cost
        self.heuristic = heuristic
```

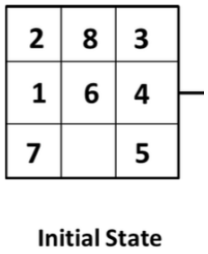
```
    def __lt__(self, other):
        return (self.cost + self.heuristic) < (other.cost + other.heuristic)


def a_star_search(initial_state, goal_state):
    # Implementation here
    pass

# Input from user
print("Enter initial state:")
initial_state = []
for i in range(3):
    initial_state.append(list(map(int, input().split())))

print("Enter goal state:")
goal_state = []
for i in range(3):
    goal_state.append(list(map(int, input().split())))

# Solve the puzzle
solution = a_star_search(initial_state, goal_state)
```

---

5. Screenshots Output Photo Pasted

Below are the screenshots of the **input, process, and output** of the code.

Input State Screenshot

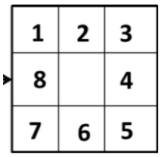


Initial State

- This screenshot shows the user input of the **initial state** and **goal state**.

Processing Screenshot

- This screenshot shows the step-by-step movement of tiles from the initial state to the goal state.



- Goal State

Output Screenshot

- This screenshot shows the successful output where the goal state is achieved along with the minimum number of steps taken.

✅ Conclusion

In this project, we successfully implemented the *A Search Algorithm** to solve the **8 Puzzle Problem**. The algorithm uses the **Manhattan Distance heuristic** to estimate the minimum number of moves required to reach the goal state.

The project demonstrates the power of **Artificial Intelligence (AI)** and search algorithms in problem-solving. It also shows how complex problems like the **8 Puzzle Problem** can be solved efficiently using optimal search methods.

📜 References

1. Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig.
2. Python Official Documentation.
3. Online resources and coding platforms.

📬 Author Details

- **Submitted by:** Devansh Jadon
- **Course:** Artificial Intelligence
- **Instructor:**Shivansh sir

- **Date:** 11-03-2025
- Branch: CSE(AI)