Offline 04 - Informed Search (A* Search)

Problem Statement

1. Given a **3×3** board with **9 tiles in total**. Among which **8 tiles** are numbered from 1 to 8 and **1 tile** is an empty tile. Initially, the board is organized in a way that all the tiles are not in their accurate position. And a **Goal state** is also given.
2. The objective is to **place the tiles to their respective positions** on the board i.e., solve the puzzle **by moving the empty space**. The empty tile can be moved into four adjacent spaces (**Right**, **Left**, **Up,** and **Down**).
3. You'll have to use the **A* Search Algorithm** to reach the **Goal state** and find the **number of moves** needed to reach the **Goal State**.

1	2	3
	4	6
7	5	8

Initial State

1	2	3
4	5	6
7	8	

Goal State

Heuristic Function

- **A* Search** is a **Heuristic Search algorithm** that uses a **Heuristic function** to know the current state of the environment.
- It **estimates the cost for each tile** in the 3x3 matrix.
- In this problem, you'll be using the **Manhattan Distance** as the **Heuristic function** to estimate the cost for each tile.
- The formula of the **Heuristic function** is

$$h(n) = Abs(row_{goalState} - row_{currentState}) + Abs(column_{goalState} - column_{currentState})$$

Cost Function

- You have to maintain another function $g(n)$ that will be **increased by 1 unit whenever you move the empty tile to any direction**.
- The **Cost Function** will be calculated using $h(n)$ and $g(n)$. And you will **choose the next state based on the minimum value** of the **Cost function**.
- The formula of the **Cost function** is

$$c(n) = f(n) + g(n)$$

Your Task

- Implement the **A* Search** algorithm as discussed in the last class and as mentioned above.
- You’ll be taking input or can hard code the **Initial State** and use the **A* Search** algorithm to reach the **Goal State**.
- Finally, print the **Goal State** and **the minimum number of moves required** to reach the **Goal State** from the given **Initial State**.

[SuAH]