

**Name:** Sai Anish Garapati

**UIN:** 650208577

## **Assignment 5:**

### **Description:**

This python file contains the A star Search implementation to solve the 15 Puzzle. The implementation is done in a generic way so that the program can solve any square matrix puzzle, but implementation assumes there is a '0' input in the matrix. This implementation also makes the assumption that the empty tile in the goal state is always at the end of the matrix (at [3, 3] in the case of 4x4 matrix puzzle).

The program implementation consists of two classes, one **PuzzleNode** class to store the state, action, parent, depth of the node in the tree and empty tile position. Another **PuzzleSolver** class to implement the A star search part, compute the heuristic function, check the goal state, and compute the child nodes. The **PuzzleSolver** class keeps the set of expanded nodes in **explored\_set** and nodes to be expanded in a **frontier** which is a Priority Queue based on min-heap which uses the sum of depth of the node and the heuristic function to set the priority within the priority queue. When the desired goal state is found, the list moves to be executed is found by backtracking from the goal state node to the root node. This 'A star search' is implemented with two heuristic functions which are **1) Misplaced tiles** and **2) Manhattan distance**. When the frontier queue becomes empty before the goal state is reached, then a solution is not possible for such an initial state.

We also print the total time taken for the algorithm to run and the number of nodes expanded and memory usage in the case a solution is found.

### **Instructions to run the code:**

This code is compiled and executed on python3 with **Python 3.8.10** version. This code uses tracemalloc to track the memory usage which is not present in python2 which could lead to an error.

The program can be run with the command **python3 650208577\_AI\_assignment\_5.py** from the command line and the user will be prompted to enter the initial state input and upon entering a square matrix, the program results in the actions taken, number of nodes expanded, time and memory usage for the execution, for both the heuristic functions misplaced tiles and manhattan distance.

### **Sample run:**

```
anish@DESKTOP-QM5TRFT: /ml  ×  +  ▾
> python3 --version
Python 3.8.10
> python3 650208577_AI_assignment_5.py
Enter initial state:1 0 2 4 5 7 3 8 9 6 11 12 13 10 14 15

Heuristic Function: Misplaced Tiles
Moves:  R D L D D R R
Number of Nodes expanded:  7
Time Taken:  0.0004830360412597656 secs
Memory used:  776B

Heuristic Function: Manhattan distance
Moves:  R D L D D R R
Number of Nodes expanded:  7
Time Taken:  0.0005943775177001953 secs
Memory used:  776B
> _
```