

Name: Sai Anish Garapati

UIN: 650208577

Assignment 6:

Description:

This python file contains the Iterative deepening 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 Iterative deepening A star search part, compute the heuristic function, check the goal state, and compute the child nodes. The **PuzzleSolver** class keeps the count of expanded nodes in **explored_nodes** and nodes to be expanded in a **frontier** which is a stack based on deque, into which nodes with $f(n)$ less than threshold are pushed. 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 becomes empty before the goal state is reached, then a solution is not possible for the current threshold and a new threshold is returned for the next iteration.

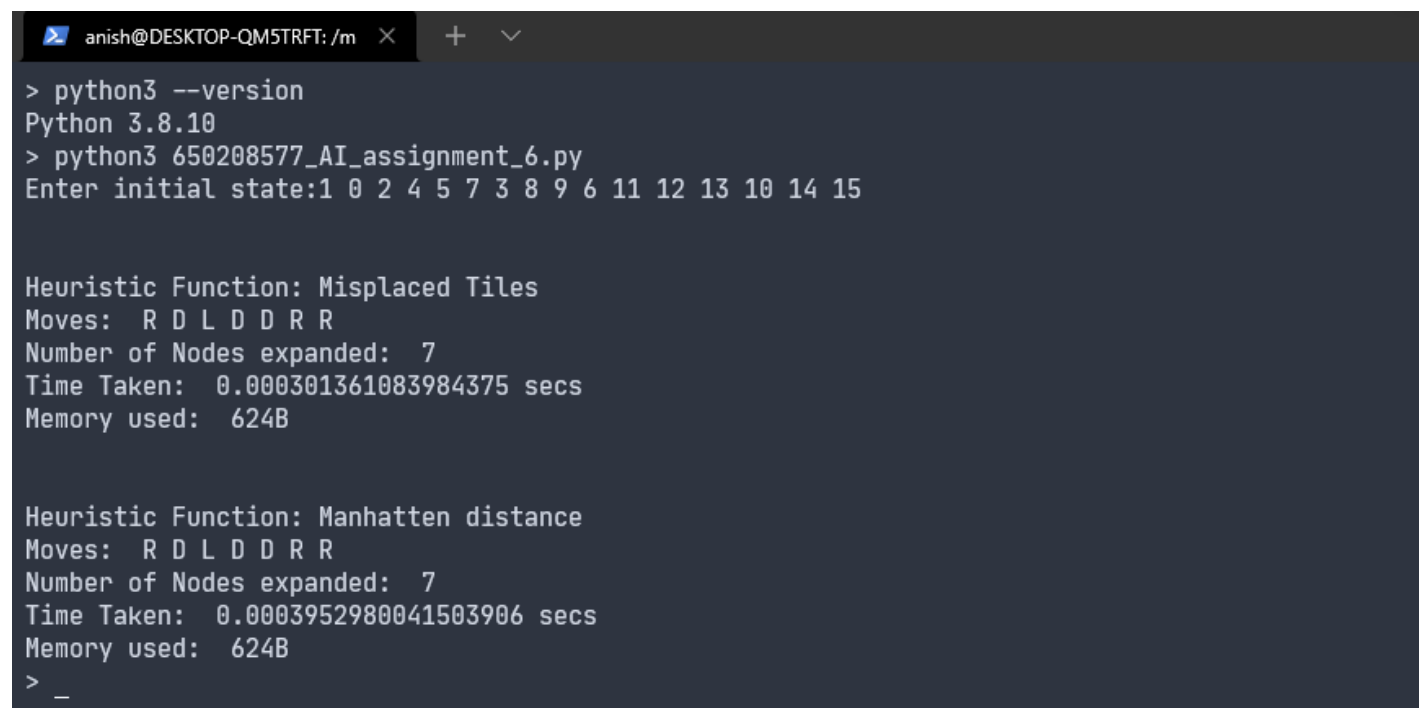
We also print the total time taken for the algorithm to run and the number of nodes expanded and memory usage in the case a solution is found.

Instructions to run the code:

This code is compiled and executed on python3 with **Python 3.8.10** version.

The program can be run with the command **python3 650208577_AI_assignment_6.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: /m
> python3 --version
Python 3.8.10
> python3 650208577_AI_assignment_6.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:  0.000301361083984375 secs
Memory used:  624B

Heuristic Function: Manhattan distance
Moves:  R D L D D R R
Number of Nodes expanded:  7
Time Taken:  0.0003952980041503906 secs
Memory used:  624B
> _
```