Assignment 1 A* algorithm for finding solution for 8 puzzle problems

This homework project is to exercise with python object-oriented programming style on implementing an 8-puzzle problem solver using A* search heuristics. There are 4 different heuristic estimation h(n). One is using number of misplaced tiles, second using Manhattan distance, third using square root Manhattan distance and fourth using max of # of misplaced tiles and Manhattan distance.

There is already a routine to check whether the puzzle is solvable by the even/odd parity calculation using inversion notion. As I explained at class.

There are several object-oriented programming concepts for Problems class, Node class that are implemented to express the problem solution search that you need to understand in order to make the Python program complete. The priority queue is to make the nodes to be explored to be sorted according to their f-evaluation function score and return the min one as the first node to be searched next.
There is also a memorize function to memorize the heuristic value of state as a look-up table so that you don’t need to calculate the redundant computing of heuristic estimation value, so you can ignore it at this point if you don’t understand.

The components you need to implement is to make the abstract part of the program realizable for 8 -puzzle with the inheritor methods attached to a problem class which consists of initial state and goal state. Make sure the program can run correctly to generate the solution sequence that move the empty tile so that the 8-puzzle can move "UP","Down","Left","RIght", from intial state to goal state.

The program is supposed to generate the solution sequences for each of the four different heuristic estimation functions given the initial state and goal state I have tentative set on (of course you can play with other initial and goal states as well when program is completed).