Programming Assignment 01: Search Algorithms (50 points)

In this assignment you will implement two search algorithms we have discussed in class: Uniform Cost Search and A\* Search. The basic algorithm can be found in the AIMA textbook or on the slides. You task is to apply the algorithm to two problems: Traveling between cities and the Sliding Puzzle. Both problems were discussed in class. For the Traveling problem, you are given a csv file that contains the distances between the cities and the straight line distance to the goal. These are the numbers found in the text book. For the Sliding Puzzle, you need to implement code for arbitrary size N, although you will only test for N=2 and 3. Lager N may run too long!

1. Use Priority Queue for the Frontier. Python has a package for priority queue, do not implement your own
2. Implement a class for the node of the search tree as described in the textbook, with state, parent, action, and path cost attributes. You will also need to add the heuristic.
3. Implement a function that generates the set of possible actions given the current state
4. All of the above should be part of a Search class.
5. Implement a separate class for the Problem, with the initial state, goal test, etc.

Since you will apply the same search algorithm to two problems, you will implement two Problem classes, but you only need one Search class to solve both problems.

Your program should output the solution as the sequence of states and actions, and the number of expanded nodes.

Start with the Travel problem as it is easier to debug. Once that problem works as expected move to the 8-Puzzle. For the 8-Puzzle, implement the Manhattan distance heuristic: h = sum of distances of each tile to its goal position. The goal state is as follows:

|  |  |  |
| --- | --- | --- |
|  | 1 | 2 |
| 3 | 4 | 5 |
| 5 | 7 | 8 |

Try to solve the following puzzles:

|  |  |  |
| --- | --- | --- |
| 5 |  | 6 |
| 1 | 2 | 8 |
| 4 | 7 | 4 |

|  |  |  |
| --- | --- | --- |
| 7 | 2 | 4 |
| 5 |  | 6 |
| 8 | 3 | 1 |

How many node expansions did it take to solve each of them?

Try to come up with your own heuristic function and report the results.

Submission

Make sure your code is fully documented and includes your name(s) in an obvious position at the top of each file you have written. Submit a brief report that shows the outputs for both problems for UC and A\* search. Discuss the results you obtained with 8-Puzzle. How long did it take to complete the UC search?

**Your code need to run at command line**