Programming Assignment- I

Search Algorithms

The 8-puzzle problem is played on a 3-by-3 grid with 8 square tiles labeled 1 through 8 and a blank tile. Your goal is to rearrange the blocks so that they are in the order shown below. You are permitted to slide blocks horizontally or vertically into the blank tile. For example, consider the given sequence:

```
6  7  1       6  7  1       6  7  1          7  1       7          1       7  8  1
8  2      =>  8     2  =>       8  2  =>  6  8  2  =>  6  8  2  =>  6          2
5  4  3       5  4  3       5  4  3       5  4  3       5  4  3       5  4  3
```

Write a program to solve the 8-puzzle problem using each of the following algorithms:

1. Depth-first search (25 points)

2. Iterative deepening search (25 points)

3. A* search using two different suitable heuristics (40 points)

10 points for analysis section described on the next page.

Your program should read the initial board configuration from any standard input and print to the standard output a sequence of board positions that solves the puzzle (only the path to goal from start), the total number of moves and the total number of search states enqueued.

For each of the above algorithms, expand the search only until depth 10, root being at depth 0. If goal state is not reached before or at depth 10, then return a failure message. Example, consider the given input and corresponding output sequence.

Input (Any random position of the tiles):

6   7   1

8   2   *

5   4   3

Output (List of states starting from input to goal state, if found):

6   7   1   (Initial input state)

8   2   *

5   4   3

```
6  7  1
8  *  2
5  4  3


6  7  1
*  8  2
5  4  3


*  7  1
6  8  2
5  4  3


7  *  1
6  8  2
5  4  3


7  8  1                                    (Goal state)
6  *  2
5  4  3
```

Number of moves = 5

Number of states enqueued = 191

Note: * represents an empty tile

What to turn in: Your code and a readme file for compiling the code. The readme file should contain the following things:

1. Instructions on how to run the program

2. Sample input and its corresponding output

3. Provide a short comparative analysis of two heuristics used for A* (10 points)

Please make sure your readme file gives clear instructions on how to run the code.

For example: `python homework1.py <algorithm_name> <input_file_path>`

where:

`algorithm_name` can take one of the following values:

- `dfs` : For running the Depth-first search algorithm

- `ids` : For running the Iterative deepening search algorithm

- `astar1` : For running the A* algorithm with heuristic 1.

- `astar2` : For running the A* algorithm with heuristic 2.

`input_file_path` : Path of the file containing the space separated input state.

For example - `6 7 1 8 2 * 5 4 3`

Note: Your submission should accept two command line parameters as shown in the example above.

Output of the program: Sequence of board positions that solves the problem, total number of moves and total number of states enqueued OR A failure message if goal state was not found before or at depth 10.

Any of the following languages can be used: - C++, Java, Python.