CS 6364: Artificial Intelligence

Homework 1: Search Algorithms (40 points)

Due: Feb 17, 2020 4pm

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 order. You are permitted to slide blocks horizontally or vertically into the blank tile. For example, consider the given sequence:

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

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

1. Breadth-first search (10 points)
2. Iterative deepening search (10 points)
3. A* search using two different heuristics (15 points)

Your program should read the initial board configuration from the 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):

*   1   3
4   2   5
7   8   6


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

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

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


1   2   3
4   *   5
7   8   6


1   2   3
4   5   *
7   8   6


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


Number of moves = 4
Number of states enqueued = 10


Note: * represents an empty tile
```

Your program should take as input one command line argument:

```
.\program <algorith_name>
```

```
where:
algorithm_name can take one of the following values:
- bfs : For running the Breadth-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.
```

Your program should output a 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.

Submit the following bundled into a single zip file via eLearning:

1. Your code files

2. A readme giving clear and precise instructions on how to run the code

3. An plaintext file showing the output of the code for the sample input given in the problem description. Additionally, describe the heuristics used for A* algorithm and analyze their performance. (5 points)