

Hamzah Dweik

1. The program can be run using the following command in the command line:

```
python3 EightPuzzle.py <algorithm_name> <input_file_path>
```

Some examples of how to run the code will be shown below as well.

2. Example input and output

Input: 6 7 1 8 2 * 5 4 3

Output:

```
• (venv) → 8puzzle python3 EightPuzzle.py dfs input_file.txt
6 7 1
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 1
6 8 2
5 4 3

7 * 1
6 8 2
5 4 3

* 7 1
6 8 2
5 4 3

7 * 1
6 8 2
5 4 3

7 8 1
6 * 2
5 4 3

Number of moves: 9
Number of states enqueued: 16
• (venv) → 8puzzle python3 EightPuzzle.py ids input_file.txt
6 7 1
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
6 * 2
5 4 3

Number of moves: 5
Number of states enqueued: 161
```

```
• (venv) → 8puzzle python3 EightPuzzle.py astar1 input_file.txt
6 7 1
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
6 * 2
5 4 3

Number of moves: 5
Number of states enqueued: 42
• (venv) → 8puzzle python3 EightPuzzle.py astar2 input_file.txt
6 7 1
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
6 * 2
5 4 3

Number of moves = 5
Number of states enqueued = 31
```

3. To begin, allow me to explain the heuristics that I decided to use in my program. The first is the Misplaced Tile Heuristic. The idea behind this heuristic is to compare the goal state and the current state to identify how many numbers are in their correct state. The second is the Euclidean Distance Heuristic. This heuristic calculates the sum total of the distances of each of the numbers in their current state compared to their location in the goal state.

Now without a doubt, I expected to see that the Euclidean Distance Heuristic would perform better because while they both encapsulate the idea of whether or not each number is in the right location, the Euclidean Distance also considers the distance of the number from its intended position. This makes it more holistic and capable of estimating the distance to the goal state. Where a Misplaced Tile Heuristic might see the following two states as equidistant from the goal state, the Euclidean Distance Heuristic would correctly identify that the first is closer to the goal state than the second. Consider:

Goal state:	First state:	Second state:
7 8 1	7 1 8	7 8 5
6 * 2	6 * 2	6 * 2
5 4 3	5 4 3	1 4 3

As I expected, our results proved this to be the case and for that reason I can almost conclusively say that Euclidean Distance was better for our application here.