

1. Professor said to include instructions in a different README file.
2. Input 1: dfs 6 7 1 8 2 * 5 4 3

Output:

Input:

6 7 1

8 2 *

5 4 3

Solution found at depth 9!

States in the path from start to goal state:

6 7 1

8 2 *

5 4 3

6 7 1

8 * 2

5 4 3

6 * 1

8 7 2

5 4 3

* 6 1

8 7 2

5 4 3

8 6 1

* 7 2

5 4 3

8 6 1

7 * 2

5 4 3

8 * 1

7 6 2

5 4 3

* 8 1

7 6 2

5 4 3

7 8 1

* 6 2

5 4 3

7 8 1

6 * 2

5 4 3

Number of moves = 9
Number of states enqueued = 2195

Input 2: ids 1 2 3 5 6 * 7 8 4

Output:

Input:

1 2 3
5 6 *
7 8 4

No solution found in the given depth limit of 0.
No solution found in the given depth limit of 1.
No solution found in the given depth limit of 2.
No solution found in the given depth limit of 3.
No solution found in the given depth limit of 4.
No solution found in the given depth limit of 5.
No solution found in the given depth limit of 6.
No solution found in the given depth limit of 7.
No solution found in the given depth limit of 8.
No solution found in the given depth limit of 9.
No solution found in the given depth limit of 10.

Input 3: astar2 * 8 1 7 5 2 4 6 3

Output:

Input:

* 8 1
7 5 2
4 6 3

Solution found at depth 6!

States in the path from start to goal state:

* 8 1
7 5 2
4 6 3

7 8 1
* 5 2

4 6 3

7 8 1

5 * 2

4 6 3

7 8 1

5 6 2

4 * 3

7 8 1

5 6 2

* 4 3

7 8 1

* 6 2

5 4 3

7 8 1

6 * 2

5 4 3

Number of moves = 6

Number of states enqueued = 17

3. The two heuristics I used for my program were the number of tiles in the wrong position (astar1) and the sum of Manhattan distances of all tiles from their goal positions (astar2). I found that heuristic 2 performs better. For example, with the input * 8 1 7 5 2 4 6 3, heuristic 1 finds the solution in 6 moves using 33 states, but heuristic 2 finds the solution in 6 moves using only 17 states. Similarly, with the input 6 7 1 8 2 * 5 4 3, heuristic 1 finds the solution in 5 moves using 18 states, but heuristic 2 finds the solution in 5 moves using only 15 states. Heuristic 2 consistently performs better because heuristic 1 can result in more equivalent function outputs than heuristic 2. There can be multiple states with the same number of misplaced tiles between the current and goal state, but it is much harder to find multiple tiles with the same Manhattan distance between the current and goal state. Since heuristic 2 is more varied with its outputs, it is easier to determine states with the best costs and to find the best path most efficiently.