

8-Puzzle Problem via A* Report

Summary

Problem

The 8-puzzle problem entails a 3x3 grid in which the numbers 0 to n are randomly shuffled within said grid. The goal is to get from some, typically random, initial state to some given goal state. To do so the agent or player must move then 0 tile around (up, left, right, or down depending on its position). Every time the 0 tile is moved it switches places with the tile in the position it is moving to. The goal is to match the goal state in as few steps as possible.

There are some cases when it is impossible for the initial state to reach the goal state. This can be tested for by checking the number of inversions the initial state has (with the goal state acting as the number system). Depending on implementation this usually causes a long search through the entire state space or an infinite loop if repeated states are not considered.

Program Structure

Running The Program

To run the program via command line simply path to the `a-star/` directory and run `python .` or `python __main__.py`.

Configure Initial and Goal States

To set the initial and goal state open the `config.yml` file and edit the matrix representations within this file. The variable `init_state` corresponds to the initial state and the variable `goal_state` corresponds to the goal state. If you wish to change the heuristic you can change the `heuristic` variable to "manhattan" or "misplaced."

`__main__.py`

The `__main__.py` file is in charge of loading the config file, checking if the given initial and goals states are valid, checking if the problem is solvable, and running

$A^*A^*A^*$. If you want to change the initial or goal state then you can do so by editing the `config.yml` (see section Configure Initial and Goal States). One function of note is the `solvable()` function which will let you know if the initial state is actually solvable. If the initial state is not solvable then you will be warned that proceeding will cause the algorithm to search the entire state space (if you still wish to continue enter "c" into the command prompt). The other functions of note are the heuristic functions `manhattan()` and `misplaced()`. The `manhattan()` is in charge of calculating the Manhattan distance of an entire state to a given goal. The `misplaced()` simply counts the number of tiles/numbers that are out of place in comparison to the goal (the max heuristic is then 9 for 3x3 when the 0 tile is included). It should be noted that `misplaced()` can take an extremely long time to solve harder problems (I do not recommend using it).

a_star.py

The `a_star.py` file contains the `AStar` and `PuzzleNode` classes. The `AStar` class is in charge of running the A^* algorithm. Once again, if the problem is not solvable the A^* algorithm will loop through the entire state space and return false. If a solution is found then it will recursively print the solution to the command line along with the nodes explored and generated and finally return true. The `PuzzleNode` class is in charge of tracking each state representation, the parent, f, g, and h scores. All the move decisions and children are generated via the `PuzzleNode` class as well.

requirements.txt

The requirements file contains all the required packages to run my code. The following command for pip should suffice (a Conda install should work as well).

```
pip install -r requirements.txt
```

Results

Manhattan

Test Number	Initial State	Goal State	Nodes Expanded	Nodes Generated (children)	Steps Needed
1	[1 2 3] [4 7 5]	[1 2 3] [4 5 6]	113	306	16

	[6 8 0]	[7 8 0]			
2	[2 8 1] [3 4 6] [7 5 0]	[3 2 1] [8 0 4] [7 5 6]	7	17	6
3	[3 5 1] [2 4 6] [7 8 0]	[3 2 1] [8 0 4] [7 5 6]	137	370	16
4	[3 5 1] [2 4 6] [7 8 0]	[5 6 0] [8 1 4] [7 2 3]	421	1126	22

Test 1

Step 0

[[1 2 3]

[4 7 5]

[6 8 0]]

$6 = 0 + 6$

Step 1

[[1 2 3]

[4 7 0]

[6 8 5]]

$8 = 1 + 7$

Step 2

[[1 2 3]

[4 0 7]

[6 8 5]]

$10 = 2 + 8$

Step 3

[[1 2 3]

[4 8 7]

[6 0 5]]

$$12 = 3 + 9$$

Step 4

[[1 2 3]

[4 8 7]

[0 6 5]]

$$12 = 4 + 8$$

Step 5

[[1 2 3]

[0 8 7]

[4 6 5]]

$$14 = 5 + 9$$

Step 6

[[1 2 3]

[8 0 7]

[4 6 5]]

$$16 = 6 + 10$$

Step 7

[[1 2 3]

[8 7 0]

[4 6 5]]

$$16 = 7 + 9$$

Step 8

[[1 2 3]

[8 7 5]

[4 6 0]]

$$16 = 8 + 8$$

Step 9

[[1 2 3]

[8 7 5]

[4 0 6]]

$$16 = 9 + 7$$

Step 10

[[1 2 3]

[8 0 5]

[4 7 6]]

$$16 = 10 + 6$$

Step 11

[[1 2 3]

[0 8 5]

[4 7 6]]

$$16 = 11 + 5$$

Step 12

[[1 2 3]

[4 8 5]

[0 7 6]]

$$16 = 12 + 4$$

Step 13

[[1 2 3]

[4 8 5]

[7 0 6]]

$$16 = 13 + 3$$

Step 14

[[1 2 3]

[4 0 5]

[7 8 6]]

$$16 = 14 + 2$$

Step 15

[[1 2 3]

[4 5 0]

[7 8 6]]

$$16 = 15 + 1$$

Step 16

[[1 2 3]

[4 5 6]
[7 8 0]]
 $16 = 16 + 0$

Test 2

Step 0
[[2 8 1]
[3 4 6]
[7 5 0]]
 $6 = 0 + 6$

Step 1
[[2 8 1]
[3 4 0]
[7 5 6]]
 $6 = 1 + 5$

Step 2
[[2 8 1]
[3 0 4]
[7 5 6]]
 $6 = 2 + 4$

Step 3
[[2 0 1]
[3 8 4]
[7 5 6]]
 $6 = 3 + 3$

Step 4
[[0 2 1]
[3 8 4]
[7 5 6]]
 $6 = 4 + 2$

Step 5
[[3 2 1]

[0 8 4]
[7 5 6]]
 $6 = 5 + 1$

Step 6
[[3 2 1]
[8 0 4]
[7 5 6]]
 $6 = 6 + 0$

Test 3

Step 0
[[3 5 1]
[2 4 6]
[7 8 0]]
 $8 = 0 + 8$

Step 1
[[3 5 1]
[2 4 0]
[7 8 6]]
 $8 = 1 + 7$

Step 2
[[3 5 1]
[2 0 4]
[7 8 6]]
 $8 = 2 + 6$

Step 3
[[3 5 1]
[2 8 4]
[7 0 6]]
 $8 = 3 + 5$

Step 4
[[3 5 1]

[2 8 4]
[0 7 6]]
 $10 = 4 + 6$

Step 5
[[3 5 1]
[0 8 4]
[2 7 6]]
 $12 = 5 + 7$

Step 6
[[3 5 1]
[8 0 4]
[2 7 6]]
 $12 = 6 + 6$

Step 7
[[3 0 1]
[8 5 4]
[2 7 6]]
 $12 = 7 + 5$

Step 8
[[0 3 1]
[8 5 4]
[2 7 6]]
 $14 = 8 + 6$

Step 9
[[8 3 1]
[0 5 4]
[2 7 6]]
 $16 = 9 + 7$

Step 10
[[8 3 1]
[2 5 4]
[0 7 6]]

$$16 = 10 + 6$$

Step 11

[[8 3 1]

[2 5 4]

[7 0 6]]

$$16 = 11 + 5$$

Step 12

[[8 3 1]

[2 0 4]

[7 5 6]]

$$16 = 12 + 4$$

Step 13

[[8 3 1]

[0 2 4]

[7 5 6]]

$$16 = 13 + 3$$

Step 14

[[0 3 1]

[8 2 4]

[7 5 6]]

$$16 = 14 + 2$$

Step 15

[[3 0 1]

[8 2 4]

[7 5 6]]

$$16 = 15 + 1$$

Step 16

[[3 2 1]

[8 0 4]

[7 5 6]]

$$16 = 16 + 0$$

Test 4

Step 0

[[3 5 1]

[2 4 6]

[7 8 0]]

$14 = 0 + 14$

Step 1

[[3 5 1]

[2 4 6]

[7 0 8]]

$16 = 1 + 15$

Step 2

[[3 5 1]

[2 0 6]

[7 4 8]]

$18 = 2 + 16$

Step 3

[[3 5 1]

[0 2 6]

[7 4 8]]

$18 = 3 + 15$

Step 4

[[0 5 1]

[3 2 6]

[7 4 8]]

$18 = 4 + 14$

Step 5

[[5 0 1]

[3 2 6]

[7 4 8]]

$18 = 5 + 13$

Step 6

[[5 1 0]

[3 2 6]

[7 4 8]]

$$18 = 6 + 12$$

Step 7

[[5 1 6]

[3 2 0]

[7 4 8]]

$$18 = 7 + 11$$

Step 8

[[5 1 6]

[3 2 8]

[7 4 0]]

$$18 = 8 + 10$$

Step 9

[[5 1 6]

[3 2 8]

[7 0 4]]

$$18 = 9 + 9$$

Step 10

[[5 1 6]

[3 0 8]

[7 2 4]]

$$18 = 10 + 8$$

Step 11

[[5 1 6]

[0 3 8]

[7 2 4]]

$$18 = 11 + 7$$

Step 12

[[5 1 6]

[7 3 8]
[0 2 4]]
 $20 = 12 + 8$

Step 13
[[5 1 6]
[7 3 8]
[2 0 4]]
 $22 = 13 + 9$

Step 14
[[5 1 6]
[7 0 8]
[2 3 4]]
 $22 = 14 + 8$

Step 15
[[5 1 6]
[7 8 0]
[2 3 4]]
 $22 = 15 + 7$

Step 16
[[5 1 6]
[7 8 4]
[2 3 0]]
 $22 = 16 + 6$

Step 17
[[5 1 6]
[7 8 4]
[2 0 3]]
 $22 = 17 + 5$

Step 18
[[5 1 6]
[7 8 4]
[0 2 3]]

$$22 = 18 + 4$$

Step 19

[[5 1 6]

[0 8 4]

[7 2 3]]

$$22 = 19 + 3$$

Step 20

[[5 1 6]

[8 0 4]

[7 2 3]]

$$22 = 20 + 2$$

Step 21

[[5 0 6]

[8 1 4]

[7 2 3]]

$$22 = 21 + 1$$

Step 22

[[5 6 0]

[8 1 4]

[7 2 3]]

$$22 = 22 + 0$$

Misplaced

Test Number	Initial State	Goal State	Nodes Expanded	Nodes Generated (children)	Steps Needed
1	[1 2 3] [4 7 5] [6 8 0]	[1 2 3] [4 5 6] [7 8 0]	598	1613	16
2	[2 8 1]	[3 2 1]	7	17	6

	[3 4 6] [7 5 0]	[8 0 4] [7 5 6]			
3	[3 5 1] [2 4 6] [7 8 0]	[3 2 1] [8 0 4] [7 5 6]	607	1744	16
4	[3 5 1] [2 4 6] [7 8 0]	[5 6 0] [8 1 4] [7 2 3]	7808	20947	22

Test 1

Step 0

[[1 2 3]

[4 7 5]

[6 8 0]]

3 = 0 + 3

Step 1

[[1 2 3]

[4 7 0]

[6 8 5]]

5 = 1 + 4

Step 2

[[1 2 3]

[4 0 7]

[6 8 5]]

6 = 2 + 4

Step 3

[[1 2 3]

[4 8 7]

[6 0 5]]

8 = 3 + 5

Step 4

[[1 2 3]

[4 8 7]
[0 6 5]]
 $9 = 4 + 5$

Step 5
[[1 2 3]
[0 8 7]
[4 6 5]]
 $11 = 5 + 6$

Step 6
[[1 2 3]
[8 0 7]
[4 6 5]]
 $12 = 6 + 6$

Step 7
[[1 2 3]
[8 7 0]
[4 6 5]]
 $13 = 7 + 6$

Step 8
[[1 2 3]
[8 7 5]
[4 6 0]]
 $13 = 8 + 5$

Step 9
[[1 2 3]
[8 7 5]
[4 0 6]]
 $15 = 9 + 6$

Step 10
[[1 2 3]
[8 0 5]
[4 7 6]]

$$16 = 10 + 6$$

Step 11

[[1 2 3]

[0 8 5]

[4 7 6]]

$$17 = 11 + 6$$

Step 12

[[1 2 3]

[4 8 5]

[0 7 6]]

$$17 = 12 + 5$$

Step 13

[[1 2 3]

[4 8 5]

[7 0 6]]

$$17 = 13 + 4$$

Step 14

[[1 2 3]

[4 0 5]

[7 8 6]]

$$17 = 14 + 3$$

Step 15

[[1 2 3]

[4 5 0]

[7 8 6]]

$$17 = 15 + 2$$

Step 16

[[1 2 3]

[4 5 6]

[7 8 0]]

$$16 = 16 + 0$$

Test 2

Step 0

[[2 8 1]

[3 4 6]

[7 5 0]]

$6 = 0 + 6$

Step 1

[[2 8 1]

[3 4 0]

[7 5 6]]

$6 = 1 + 5$

Step 2

[[2 8 1]

[3 0 4]

[7 5 6]]

$5 = 2 + 3$

Step 3

[[2 0 1]

[3 8 4]

[7 5 6]]

$7 = 3 + 4$

Step 4

[[0 2 1]

[3 8 4]

[7 5 6]]

$7 = 4 + 3$

Step 5

[[3 2 1]

[0 8 4]

[7 5 6]]

$7 = 5 + 2$

Step 6

[[3 2 1]

[8 0 4]

[7 5 6]]

$6 = 6 + 0$

Test 3

Step 0

[[3 5 1]

[2 4 6]

[7 8 0]]

$6 = 0 + 6$

Step 1

[[3 5 1]

[2 4 0]

[7 8 6]]

$6 = 1 + 5$

Step 2

[[3 5 1]

[2 0 4]

[7 8 6]]

$5 = 2 + 3$

Step 3

[[3 5 1]

[2 8 4]

[7 0 6]]

$7 = 3 + 4$

Step 4

[[3 5 1]

[2 8 4]

[0 7 6]]

$9 = 4 + 5$

Step 5

[[3 5 1]

[0 8 4]

[2 7 6]]

$$10 = 5 + 5$$

Step 6

[[3 5 1]

[8 0 4]

[2 7 6]]

$$9 = 6 + 3$$

Step 7

[[3 0 1]

[8 5 4]

[2 7 6]]

$$11 = 7 + 4$$

Step 8

[[0 3 1]

[8 5 4]

[2 7 6]]

$$13 = 8 + 5$$

Step 9

[[8 3 1]

[0 5 4]

[2 7 6]]

$$15 = 9 + 6$$

Step 10

[[8 3 1]

[2 5 4]

[0 7 6]]

$$16 = 10 + 6$$

Step 11

[[8 3 1]

[2 5 4]
[7 0 6]]
 $16 = 11 + 5$

Step 12
[[8 3 1]
[2 0 4]
[7 5 6]]
 $15 = 12 + 3$

Step 13
[[8 3 1]
[0 2 4]
[7 5 6]]
 $17 = 13 + 4$

Step 14
[[0 3 1]
[8 2 4]
[7 5 6]]
 $17 = 14 + 3$

Step 15
[[3 0 1]
[8 2 4]
[7 5 6]]
 $17 = 15 + 2$

Step 16
[[3 2 1]
[8 0 4]
[7 5 6]]
 $16 = 16 + 0$

Test 4

Step 0
[[3 5 1]

[2 4 6]
[7 8 0]]
 $8 = 0 + 8$

Step 1
[[3 5 1]
[2 4 6]
[7 0 8]]
 $9 = 1 + 8$

Step 2
[[3 5 1]
[2 0 6]
[7 4 8]]
 $10 = 2 + 8$

Step 3
[[3 5 1]
[0 2 6]
[7 4 8]]
 $11 = 3 + 8$

Step 4
[[0 5 1]
[3 2 6]
[7 4 8]]
 $12 = 4 + 8$

Step 5
[[5 0 1]
[3 2 6]
[7 4 8]]
 $12 = 5 + 7$

Step 6
[[5 1 0]
[3 2 6]
[7 4 8]]

$$12 = 6 + 6$$

Step 7

[[5 1 6]

[3 2 0]

[7 4 8]]

$$14 = 7 + 7$$

Step 8

[[5 1 6]

[3 2 8]

[7 4 0]]

$$15 = 8 + 7$$

Step 9

[[5 1 6]

[3 2 8]

[7 0 4]]

$$16 = 9 + 7$$

Step 10

[[5 1 6]

[3 0 8]

[7 2 4]]

$$16 = 10 + 6$$

Step 11

[[5 1 6]

[0 3 8]

[7 2 4]]

$$17 = 11 + 6$$

Step 12

[[5 1 6]

[7 3 8]

[0 2 4]]

$$19 = 12 + 7$$

Step 13

[[5 1 6]

[7 3 8]

[2 0 4]]

$$21 = 13 + 8$$

Step 14

[[5 1 6]

[7 0 8]

[2 3 4]]

$$22 = 14 + 8$$

Step 15

[[5 1 6]

[7 8 0]

[2 3 4]]

$$23 = 15 + 8$$

Step 16

[[5 1 6]

[7 8 4]

[2 3 0]]

$$23 = 16 + 7$$

Step 17

[[5 1 6]

[7 8 4]

[2 0 3]]

$$23 = 17 + 6$$

Step 18

[[5 1 6]

[7 8 4]

[0 2 3]]

$$23 = 18 + 5$$

Step 19

[[5 1 6]

[0 8 4]

[7 2 3]]

$$23 = 19 + 4$$

Step 20

[[5 1 6]

[8 0 4]

[7 2 3]]

$$23 = 20 + 3$$

Step 21

[[5 0 6]

[8 1 4]

[7 2 3]]

$$23 = 21 + 2$$

Step 22

[[5 6 0]

[8 1 4]

[7 2 3]]

$$22 = 22 + 0$$