

Puzzle A. (should be very fast)

python3 UCS.py EightPuzzle '[[3,0,1],[6,4,2],[7,8,5]]'

Initial state as given on the command line: [[3,0,1],[6,4,2],[7,8,5]]

Welcome to UCS.

Initial State:

```
[[3, 0, 1]
 [6, 4, 2]
 [7, 8, 5]]
```

You've got all eight straight. Great!

Solution path:

```
[[3, 0, 1]
 [6, 4, 2]
 [7, 8, 5]]
```

```
[[3, 1, 0]
 [6, 4, 2]
 [7, 8, 5]]
```

```
[[3, 1, 2]
 [6, 4, 0]
 [7, 8, 5]]
```

```
[[3, 1, 2]
 [6, 4, 5]
 [7, 8, 0]]
```

```
[[3, 1, 2]
 [6, 4, 5]
 [7, 0, 8]]
```

```
[[3, 1, 2]
 [6, 4, 5]
 [0, 7, 8]]
```

```
[[3, 1, 2]
 [0, 4, 5]
 [6, 7, 8]]
```

```
[[0, 1, 2]
 [3, 4, 5]
 [6, 7, 8]]
```

Length of solution path found: 7 edges

Total cost of solution path found: 7.0

166 states expanded.

MAX_OPEN_LENGTH = 101

Puzzle B. (should not take long)

python3 UCS.py EightPuzzle '[[3,1,2],[6,8,7],
[5,4,0]]'

Welcome to UCS.
Initial State:

```
[[3, 1, 2]
 [6, 8, 7]
 [5, 4, 0]]
```

You've got all eight straight. Great!
Solution path:

```
[[3, 1, 2]
 [6, 8, 7]
 [5, 4, 0]]
```

```
[[3, 1, 2]
 [6, 8, 0]
 [5, 4, 7]]
```

```
[[3, 1, 2]
 [6, 0, 8]
 [5, 4, 7]]
```

```
[[3, 1, 2]
 [6, 4, 8]
 [5, 0, 7]]
```

```
[[3, 1, 2]
 [6, 4, 8]
 [0, 5, 7]]
```

```
[[3, 1, 2]
 [0, 4, 8]
 [6, 5, 7]]
```

```
[[3, 1, 2]
 [4, 0, 8]
 [6, 5, 7]]
```

```
[[3, 1, 2]
 [4, 5, 8]
 [6, 0, 7]]
```

```
[[3, 1, 2]
```

```
[4, 5, 8]  
[6, 7, 0]]
```

```
[[3, 1, 2]  
[4, 5, 0]  
[6, 7, 8]]
```

```
[[3, 1, 2]  
[4, 0, 5]  
[6, 7, 8]]
```

```
[[3, 1, 2]  
[0, 4, 5]  
[6, 7, 8]]
```

```
[[0, 1, 2]  
[3, 4, 5]  
[6, 7, 8]]
```

Length of solution path found: 12 edges
Total cost of solution path found: 12.0
1490 states expanded.
MAX_OPEN_LENGTH = 898

Puzzle C. (May take a few minutes)

**python3 UCS.py EightPuzzle '[[4,5,0],[1,2,8],
[3,7,6]]'**

Initial state as given on the command line: [[3,1,2],[6,8,7],[5,4,0]]

Welcome to UCS.

Initial State:

```
[[3, 1, 2]
 [6, 8, 7]
 [5, 4, 0]]
```

You've got all eight straight. Great!

Solution path:

```
[[4, 5, 0]
 [1, 2, 8]
 [3, 7, 6]]
```

```
[[4, 0, 5]
 [1, 2, 8]
 [3, 7, 6]]
```

```
[[4, 2, 5]
 [1, 0, 8]
 [3, 7, 6]]
```

```
[[4, 2, 5]
 [1, 7, 8]
 [3, 0, 6]]
```

```
[[4, 2, 5]
 [1, 7, 8]
 [3, 6, 0]]
```

```
[[4, 2, 5]
 [1, 7, 0]
 [3, 6, 8]]
```

```
[[4, 2, 0]
 [1, 7, 5]
 [3, 6, 8]]
```

```
[[4, 0, 2]
 [1, 7, 5]
 [3, 6, 8]]
```

```
[[0, 4, 2]
 [1, 7, 5]
 [3, 6, 8]]
```

```
[[1, 4, 2]
 [0, 7, 5]
 [3, 6, 8]]
```

```
[[1, 4, 2]
 [3, 7, 5]
 [0, 6, 8]]
```

```
[[1, 4, 2]
 [3, 7, 5]
 [6, 0, 8]]
```

```
[[1, 4, 2]
 [3, 0, 5]
 [6, 7, 8]]
```

```
[[1, 0, 2]
 [3, 4, 5]
 [6, 7, 8]]
```

```
[[0, 1, 2]
 [3, 4, 5]
 [6, 7, 8]]
```

Length of solution path found: 14 edges

Total cost of solution path found: 14.0

4070 states expanded.

MAX_OPEN_LENGTH = 2290

Puzzle D. (May take several minutes)

python3 UCS.py EightPuzzle '[[0,8,2],[1,7,4],[3,6,5]]'

Initial state as given on the command line: [[0,8,2],[1,7,4],[3,6,5]]

Welcome to UCS.

Initial State:

```
[[0, 8, 2]
 [1, 7, 4]
 [3, 6, 5]]
```

You've got all eight straight. Great!

Solution path:

```
[[0, 8, 2]
 [1, 7, 4]
 [3, 6, 5]]
```

```
[[1, 8, 2]
 [0, 7, 4]
 [3, 6, 5]]
```

```
[[1, 8, 2]
 [3, 7, 4]
 [0, 6, 5]]
```

```
[[1, 8, 2]
 [3, 7, 4]
 [6, 0, 5]]
```

```
[[1, 8, 2]
 [3, 0, 4]
 [6, 7, 5]]
```

```
[[1, 0, 2]
 [3, 8, 4]
 [6, 7, 5]]
```

```
[[0, 1, 2]
 [3, 8, 4]
 [6, 7, 5]]
```

```
[[3, 1, 2]
 [0, 8, 4]
 [6, 7, 5]]
```

```
[[3, 1, 2]
 [6, 8, 4]
 [0, 7, 5]]
```

```
[[3, 1, 2]
 [6, 8, 4]
 [7, 0, 5]]
```

```
[[3, 1, 2]
 [6, 0, 4]
 [7, 8, 5]]
```

```
[[3, 1, 2]
 [6, 4, 0]
 [7, 8, 5]]
```

```
[[3, 1, 2]
 [6, 4, 5]
 [7, 8, 0]]
```

```
[[3, 1, 2]
 [6, 4, 5]
 [7, 0, 8]]
```

```
[[3, 1, 2]
 [6, 4, 5]
 [0, 7, 8]]
```

```
[[3, 1, 2]
 [0, 4, 5]
 [6, 7, 8]]
```

```
[[0, 1, 2]
 [3, 4, 5]
 [6, 7, 8]]
```

Length of solution path found: 16 edges

Total cost of solution path found: 16.0

7982 states expanded.

MAX_OPEN_LENGTH = 4700

Heuristics for the Eight Puzzle

Eight Puzzle With Hamming:

Welcome to Astar.
Initial State:

```
[[3, 1, 2]
 [0, 5, 8]
 [4, 6, 7]]
len(OPEN)=1; len(CLOSED)=0; COUNT = 0
len(OPEN)=3; len(CLOSED)=1; COUNT = 1
len(OPEN)=3; len(CLOSED)=2; COUNT = 2
len(OPEN)=3; len(CLOSED)=3; COUNT = 3
len(OPEN)=5; len(CLOSED)=4; COUNT = 4
len(OPEN)=6; len(CLOSED)=5; COUNT = 5
len(OPEN)=6; len(CLOSED)=6; COUNT = 6
len(OPEN)=7; len(CLOSED)=7; COUNT = 7
len(OPEN)=9; len(CLOSED)=8; COUNT = 8
len(OPEN)=10; len(CLOSED)=9; COUNT = 9
You've got all eight straight. Great!
Solution path:
```

```
[[3, 1, 2]
 [0, 5, 8]
 [4, 6, 7]]
```

```
[[3, 1, 2]
 [4, 5, 8]
 [0, 6, 7]]
```

```
[[3, 1, 2]
 [4, 5, 8]
 [6, 0, 7]]
```

```
[[3, 1, 2]
 [4, 5, 8]
 [6, 7, 0]]
```

```
[[3, 1, 2]
 [4, 5, 0]
 [6, 7, 8]]
```

```
[[3, 1, 2]
 [4, 0, 5]
 [6, 7, 8]]
```

```
[[3, 1, 2]
```

```

[0, 4, 5]
[6, 7, 8]]

[[0, 1, 2]
 [3, 4, 5]
 [6, 7, 8]]
Length of solution path found: 7 edges
Total cost of solution path found: 7.0
9 states expanded.
MAX_OPEN_LENGTH = 10

```

Eight Puzzle With Manhattan

Welcome to Astar.
Initial State:

```

[[3, 1, 2]
 [0, 5, 8]
 [4, 6, 7]]
len(OPEN)=1; len(CLOSED)=0; COUNT = 0
len(OPEN)=3; len(CLOSED)=1; COUNT = 1
len(OPEN)=3; len(CLOSED)=2; COUNT = 2
len(OPEN)=3; len(CLOSED)=3; COUNT = 3
len(OPEN)=5; len(CLOSED)=4; COUNT = 4
len(OPEN)=6; len(CLOSED)=5; COUNT = 5
len(OPEN)=7; len(CLOSED)=6; COUNT = 6
len(OPEN)=7; len(CLOSED)=7; COUNT = 7
len(OPEN)=8; len(CLOSED)=8; COUNT = 8
len(OPEN)=10; len(CLOSED)=9; COUNT = 9
len(OPEN)=11; len(CLOSED)=10; COUNT = 10
You've got all eight straight. Great!
Solution path:

```

```

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

```

```

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

```

```

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

```

```

[[3, 1, 2]
 [4, 5, 8]

```

[6, 7, 0]]

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

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

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

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

Length of solution path found: 7 edges
Total cost of solution path found: 7.0
10 states expanded.
MAX_OPEN_LENGTH = 11