

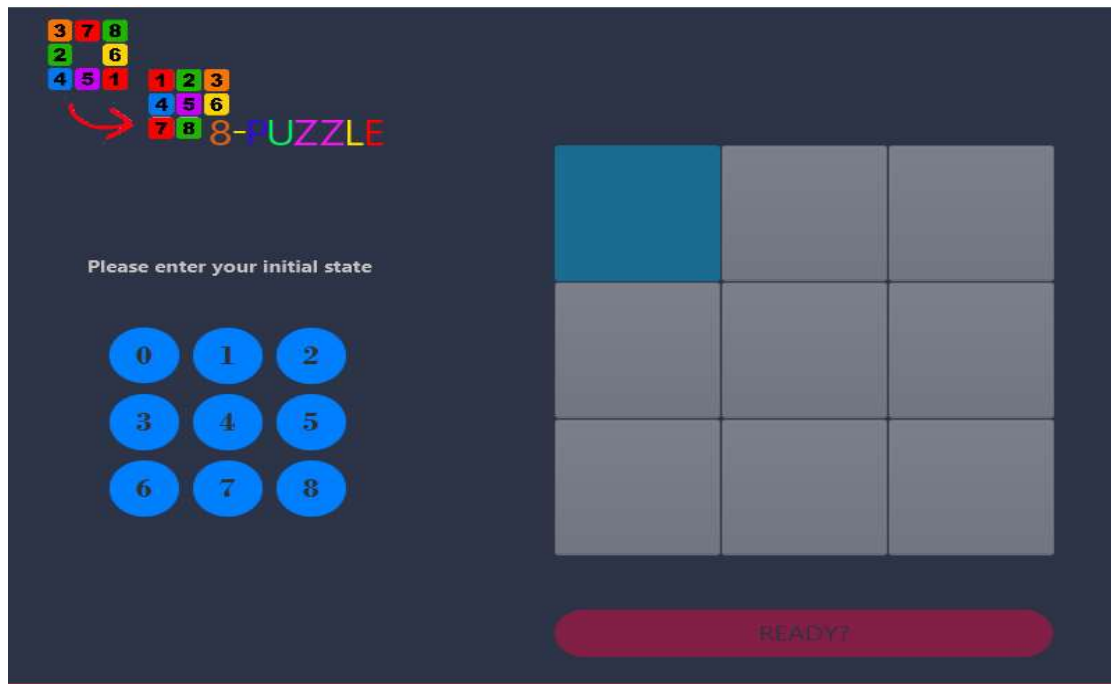
Artificial Intelligence

8-PUZZLE SOLVER

Mohamed Hany

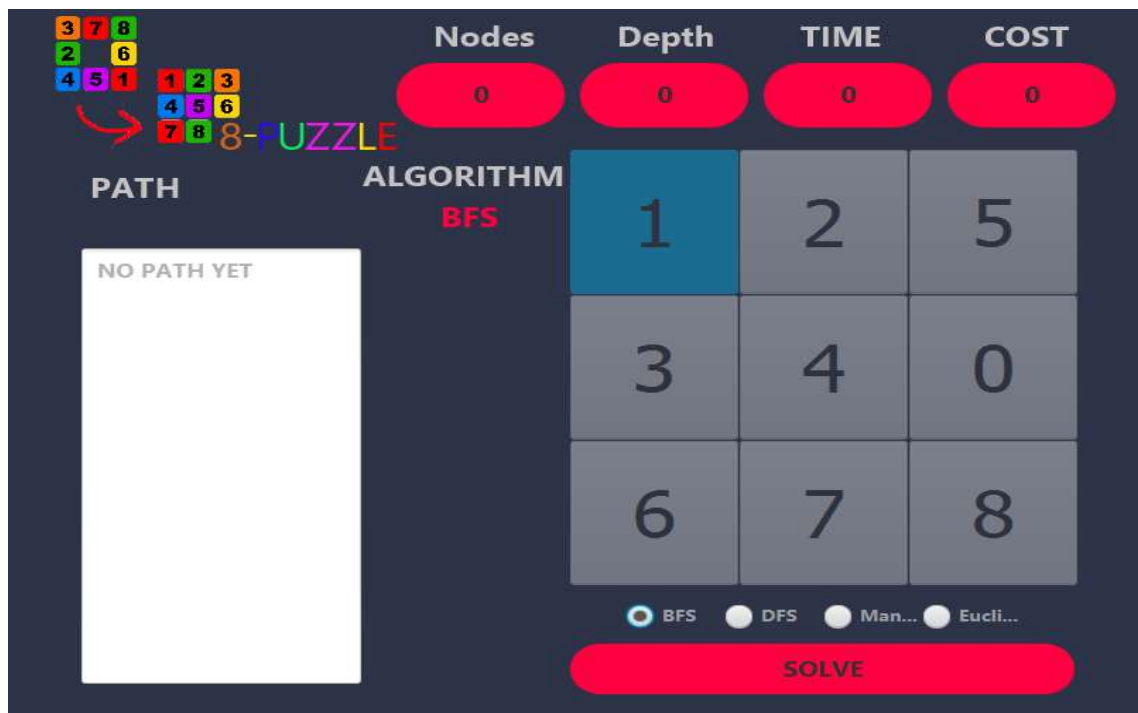
4580

How to use the app:



After entering the initial state the ready button will be enabled, press it.

It will take you to the second screen:



Choose your algorithm, press solve and let it go. (it may take too long)

Data Structures:

For the explored nodes: queue.

For the frontier in BFS: queue.

For the frontier in DFS: stack.

For the frontier in A*: priority queue.

Assumptions:

The goal : {0,1,2,3,4,5,6,7,8}

The moves: Up >> Down >> Left >> Right

Sample run:

The initial state : {1,2,5,3,4,0,6,7,8}.

The screenshot displays an 8-puzzle solver interface. At the top left, the initial state is shown as a 3x3 grid of colored tiles: top row (3, 7, 8), middle row (2, 6, 1), and bottom row (4, 5, 1). A red arrow points from the tile with '1' in the middle-right position to its new position in the bottom-right position. Below this, the text '8-PUZZLE' is displayed. To the right of the grid, four red buttons show search statistics: 'Nodes' (11), 'Depth' (3), 'TIME' (0), and 'COST' (3). Below the grid, a 'PATH' section shows a list of moves: 'up', 'left', and 'left'. To the right of the path, the 'ALGORITHM' is set to 'BFS'. At the bottom, there are four radio buttons for selecting the algorithm: 'BFS' (selected), 'DFS', 'Man...', and 'Eucli...'. A large red 'SOLVE' button is at the bottom right.

Nodes	Depth	TIME	COST
11	3	0	3

PATH

- up
- left
- left

ALGORITHM

BFS

BFS DFS Man... Eucli...

SOLVE

3

7

8

2

6

4

5

1

1

2

3

4

5

6

7

8

8-PUZZLE

Nodes

439

Depth

431

TIME

0

COST

431

PATH

left
left
down
right
right
up
left
left
down
right
right
up
left

ALGORITHM

DFS

1

2

5

3

4

0

6

7

8

☐ BFS

☒ DFS

☐ Man...

☐ Eucli...

SOLVE

3

7

8

2

6

4

5

1

1

2

3

4

5

6

7

8

8-PUZZLE

Nodes

6

Depth

3

TIME

0

COST

3

PATH

up
left
left

ALGORITHM

A*
Manhattan

1

2

5

3

4

0

6

7

8

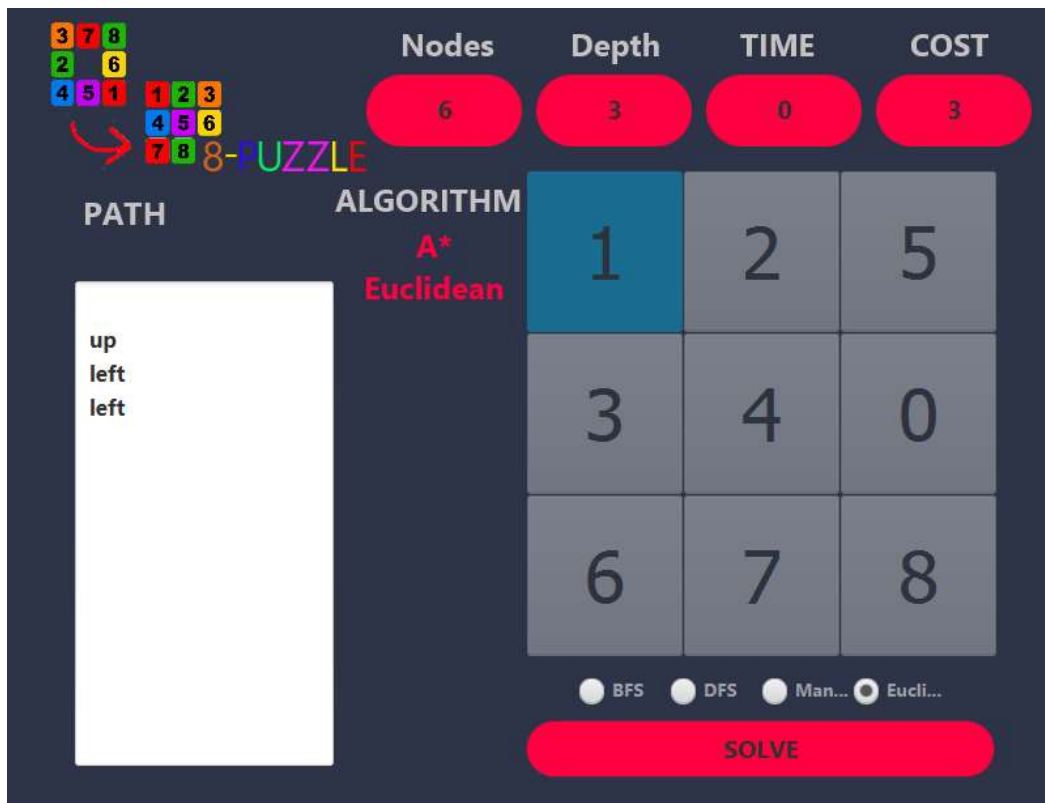
☐ BFS

☐ DFS

☒ Man...

☐ Eucli...

SOLVE



Conclusion:

Using "Manhattan" as heuristic takes less runtime than "Euclidean" and expands less nodes.