

## Assignment 1 - Using Informed and Uninformed Search Algorithms to Solve 8-Puzzle

Omar Mohamed AbdelBaset Mohamed

# 44

### Heuristics:

- The Manhattan Distance Heuristic is not admissible

Consider state

3	1	2
6	4	5
	7	8

3	1	2
	4	5
6	7	8

	1	2
3	4	5
6	7	8

The total cost = 2 moves

The manhatan distance =  $2 + 0 + 0 + 1 + 0 + 0 + 2 + 0 + 0 + 0$

=> Over estimates a solution => not admissible

- The Eucledian Distance Heuristic is more admissible than the Manhattan distance as the result is always smaller or equal to the other's.

Example where the Manhattan distance fails:

5	7	0
6	8	1
3	2	4

A\* with manhatan => cost = 24

A\* with Eucledian => cost = 22

### **Used Data Structures:**

- PuzzleBoard: a 2d tuple representing the puzzle.
- GameState(PuzzleBoard, move, cost, depth): Holds information about the current game state, passing this to the searching algorithms along with the callback needed to evaluate Neighbouring states from current state.
- Queue: Used in the BFS searching algorithm
- Priority Queue: Used in the A\* searching algorithm, using the heuristic function(state) + state.cost to compare between the states in the queue.
- GameStateWrapper: a wrapper class around the GameState class to add functionality to evaluate the selected heuristic function upon comparing it to another state. Used in the priority queue in the A\* algorithm.

## Samples Runs

### Using BFS:

Finding a path to goal using <class 'informed\_search\_algorithms.Bfs'> None for board:

3	1	2
4	5	
6	7	8

Move: Up

3	1	2
4		5
6	7	8

Move: Up

3	1	2
	4	5
6	7	8

Move: Left

	1	2
3	4	5
6	7	8

Result of using <class 'informed\_search\_algorithms.Bfs'> None

Total path cost: 3

Depth: 3

Expanded nodes count: 18

Running time: 0.0024619102478027344

Running time:  
Finding a path to goal using <class 'informed\_search\_algorithms.Bfs'> None for board:

3	1	2
6	4	5
7	8	

Move: Up

3	1	2
6	4	5
7		8

Move: Up

3	1	2
6	4	5
	7	8

Move: Left

3	1	2
	4	5
6	7	8

Move: Left

	1	2
3	4	5
6	7	8

Result of using <class 'informed\_search\_algorithms.Bfs'> None  
Total path cost: 4  
Depth: 4  
Expanded nodes count: 30  
Running time: 0.0011854171752929688  
Finding a path to goal using <class 'informed\_search\_algorithms.Bfs'> None for board:

Finding a path to goal using <class 'informed\_search\_algorithms.Bfs'> None for board:

3		2
6	1	5
7	4	8

Move: Right

3	1	2
6		5
7	4	8

Move: Right

3	1	2
6	4	5
7		8

Move: Up

3	1	2
6	4	5
	7	8

Move: Left

3	1	2
	4	5
6	7	8

Move: Left

	1	2
3	4	5
6	7	8

Result of using <class 'informed\_search\_algorithms.Bfs'> None  
Total path cost: 5  
Depth: 5  
Expanded nodes count: 34  
Running time: 0.0014576911926269531

## Using DFS:

It was mostly stack overflows as the path was so deep and python doesn't have many optimization to recursion like tail recursion.

```
Finding a path to goal using <class 'informed_search_algorithms.Dfs'> None for board:
```

3		2
6	1	5
7	4	8

```
Traceback (most recent call last):
```

```
File "index.py", line 96, in <module>
    evaluate(board, Dfs)
File "index.py", line 50, in evaluate
    init_state, puzzle.GOAL_STATE, puzzle.generate_neighbours)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 74, in solve
    set(), [], 0)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 62, in _solve
    parents, expanded_nodes_cnt)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 62, in _solve
    parents, expanded_nodes_cnt)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 62, in _solve
    parents, expanded_nodes_cnt)
[Previous line repeated 985 more times]
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 59, in _solve
    for neighbour in reversed(get_neighbours_fn(current_state)):
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 119, in generate_neighbours
    move=move[2]) for move in _generate_valid_moves(state.board)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 119, in <listcomp>
    move=move[2]) for move in _generate_valid_moves(state.board)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 105, in _transition
    }) for row in range(N)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 105, in <listcomp>
    }) for row in range(N)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 104, in <listcomp>
    for col in range(N)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 89, in _transition_get_item
    if (row, col) == empty_space_pos:
RecursionError: maximum recursion depth exceeded in comparison
```

WARNING: Ignoring invalid data fields and extra information.

Finding a path to goal using <class 'informed\_search\_algorithms.Dfs'> None for board:

3	1	2
6	4	5
7	8	

Traceback (most recent call last):

```
File "index.py", line 96, in <module>
    evaluate(board, Dfs)
File "index.py", line 50, in evaluate
    init_state, puzzle.GOAL_STATE, puzzle.generate_neighbours)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 74, in solve
    set(), [], 0)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 62, in _solve
    parents, expanded_nodes_cnt)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 62, in _solve
    parents, expanded_nodes_cnt)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 62, in _solve
    parents, expanded_nodes_cnt)
[Previous line repeated 985 more times]
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 59, in _solve
    for neighbour in reversed(get_neighbours_fn(current state)):
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 119, in generate_neighbours
    move=move[2]) for move in _generate_valid_moves(state.board)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 119, in <listcomp>
    move=move[2]) for move in _generate_valid_moves(state.board)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 105, in _transition
    ]) for row in range(N)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 105, in <listcomp>
    ]) for row in range(N)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 104, in <listcomp>
    for col in range(N)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 89, in _transition_get_item
    if (row, col) == empty_space_pos:
RecursionError: maximum recursion depth exceeded in comparison
```



Finding a path to goal using <class 'informed\_search\_algorithms.Dfs'> None for board:

3	1	2
4	5	
6	7	8

Traceback (most recent call last):

```
File "index.py", line 96, in <module>
    evaluate(board, Dfs)
File "index.py", line 50, in evaluate
    init_state, puzzle.GOAL_STATE, puzzle.generate_neighbours)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 74, in solve
    set(), [], 0)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 62, in _solve
    parents, expanded_nodes_cnt)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 62, in _solve
    parents, expanded_nodes_cnt)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 62, in _solve
    parents, expanded_nodes_cnt)
[Previous line repeated 985 more times]
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/informed_search_algorithms.py", line 59, in _solve
    for neighbour in reversed(get_neighbours_fn(current_state)):
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 119, in generate_neighbours
    move=move[2]) for move in _generate_valid_moves(state.board)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 119, in <listcomp>
    move=move[2]) for move in _generate_valid_moves(state.board)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 105, in _transition
    ]) for row in range(N)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 105, in <listcomp>
    ]) for row in range(N)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 104, in <listcomp>
    for col in range(N)
File "/home/vontman/Dropbox/projects/8_Puzzle_Ai/puzzle.py", line 89, in _transition_get_item
    if (row, col) == empty_space_pos:
RecursionError: maximum recursion depth exceeded in comparison
```

# Using A\* with Manhattan Distance:

```
Finding a path to goal using <class 'uninformed_search_algorithms.AStar'> <function manhatanHeuristic at 0x7f833e13c9d8> for board:
| 3 | 1 | 2 |
| 4 | 5 |   |
| 6 | 7 | 8 |
|   |   |   |

Move: Up
| 3 | 1 | 2 |
| 4 |   | 5 |
| 6 | 7 | 8 |
|   |   |   |

Move: Up
| 3 | 1 | 2 |
|   | 4 | 5 |
| 6 | 7 | 8 |
|   |   |   |

Move: Left
|   | 1 | 2 |
| 3 | 4 | 5 |
| 6 | 7 | 8 |
|   |   |   |

Result of using <class 'uninformed_search_algorithms.AStar'> <function manhatanHeuristic at 0x7f833e13c9d8>
Total path cost: 3
Depth: 3
Expanded nodes count: 4
Running time: 0.00431513786315918
```

Finding a path to goal using <class 'uninformed\_search\_algorithms.AStar'> <function manhatanHeuristic at 0x7f3f08e379d8> for board:

3	1	2
6	4	5
7	8	

Move: Up

3	1	2
6	4	5
7		8

Move: Up

3	1	2
6	4	5
	7	8

Move: Left

3	1	2
	4	5
6	7	8

Move: Left

	1	2
3	4	5
6	7	8

Result of using <class 'uninformed\_search\_algorithms.AStar'> <function manhatanHeuristic at 0x7f3f08e379d8>  
Total path cost: 4  
Depth: 4  
Expanded nodes count: 5  
Running time: 0.003524303436279297

Finding a path to goal using <class 'uninformed\_search\_algorithms.AStar'> <function manhatanHeuristic at 0x7fc07a7b19d8> for board:

3		2
6	1	5
7	4	8

Move: Right

3	1	2
6		5
7	4	8

Move: Right

3	1	2
6	4	5
7		8

Move: Up

3	1	2
6	4	5
	7	8

Move: Left

3	1	2
	4	5
6	7	8

Move: Left

	1	2
3	4	5
6	7	8

Result of using <class 'uninformed\_search\_algorithms.AStar'> <function manhatanHeuristic at 0x7fc07a7b19d8>  
Total path cost: 5  
Depth: 5  
Expanded nodes count: 14  
Running time: 0.014968156814575195

## Using A\* with Euclidian Distance:

Finding a path to goal using <class 'uninformed\_search\_algorithms.AStar'> <function euclidianHeuristic at 0x7f833e13c950> for board:

3	1	2
4	5	
6	7	8

Move: Up

3	1	2
4		5
6	7	8

Move: Up

3	1	2
	4	5
6	7	8

Move: Left

	1	2
3	4	5
6	7	8

Result of using <class 'uninformed\_search\_algorithms.AStar'> <function euclidianHeuristic at 0x7f833e13c950>

Total path cost: 3

Depth: 3

Expanded nodes count: 4

Running time: 0.00457763671875

Finding a path to goal using <class 'uninformed\_search\_algorithms.AStar'> <function eucledianHeuristic at 0x7f3f08e37950> for board:

3	1	2	
6	4	5	
7	8		

Move: Up

3	1	2	
6	4	5	
7		8	

Move: Up

3	1	2	
6	4	5	
	7	8	

Move: Left

3	1	2	
	4	5	
6	7	8	

Move: Left

	1	2	
3	4	5	
6	7	8	

Result of using <class 'uninformed\_search\_algorithms.AStar'> <function eucledianHeuristic at 0x7f3f08e37950>

Total path cost: 4

Depth: 4

Expanded nodes count: 10

Running time: 0.009241819381713867

Finding a path to goal using <class 'uninformed\_search\_algorithms.Bfs'> <function eucledianHeuristic at 0x7f3f08e37950> for board:

Finding a path to goal using <class 'uninformed\_search\_algorithms.AStar'> <function eucledianHeuristic at 0x7fc07a7b1950> for board:

3		2
6	1	5
7	4	8

Move: Right

3	1	2
6		5
7	4	8

Move: Right

3	1	2
6	4	5
7		8

Move: Up

3	1	2
6	4	5
	7	8

Move: Left

3	1	2
	4	5
6	7	8

Move: Left

	1	2
3	4	5
6	7	8

Result of using <class 'uninformed\_search\_algorithms.AStar'> <function eucledianHeuristic at 0x7fc07a7b1950>  
Total path cost: 5  
Depth: 5  
Expanded nodes count: 14  
Running time: 0.005842924118041992

## Notes:

- Currently to run the code, `python3 index.py` .
- Added docstrings for the main functions.
- Using mypy with python 3 to add static typing.
- Applying some functional programming concepts so I am sorry for the complication.
- The code could use some refactoring here and there.
- Python is not a great language to write templated and well structured code.