Arman Akbari – 610399100 second assignment - AI

Assignment description:

The 8-puzzle game is a sliding puzzle that consists of a 3x3 grid with 8 numbered tiles and one blank space. The goal is to rearrange the tiles from a start state to a goal state by sliding the tiles into the blank space.

The following are the components for solving the 8-puzzle game with search algorithms:

1. **States**: The state of the puzzle at a given moment, represented by an arrangement of the tiles on the grid.
2. **Actions**: The possible moves of the blank space, which are either up, down, left, or right.
3. **Goal Test**: A function that determines if the current state is the goal state.
4. **Path Cost**: A value assigned to each step in the search, representing the cost of getting from one state to another. In the 8-puzzle game, the path cost is usually assigned a value of 1 for each move.
5. **Transition Model**: A function that takes the current state and an action, and returns the next state after that action is executed.

The Manhattan distance heuristic:

The Manhattan distance is the sum of the distances of each tile from its goal position, measured along the rows and columns. It is an admissible and consistent heuristic, which means it never overestimates the true distance and satisfies the triangle inequality. This makes it well-suited for guiding the search towards the goal.

1. **Breadth-First Search (BFS):** BFS is guaranteed to find the optimal solution in the least number of moves, but its time and space complexity can be very high, as it needs to keep track of all possible states at every depth level.
2. **Depth-First Search (DFS):** DFS has low time and space complexity, but it may not find the optimal solution and can get stuck in an infinite loop if the depth limit is not set properly.
3. **Iterative Deepening Search (IDS):** IDS combines the best of DFS and BFS by gradually increasing the depth limit to ensure optimality while avoiding the infinite loop problem. However, it still has a high time complexity, especially for deep states.
4. **Uniform Cost Search (UCS):** UCS is similar to BFS in terms of finding the optimal solution, but it prioritizes states based on their cost rather than their depth level. It is useful for solving problems with non-uniform step costs.
5. **A* Search:** A* search uses a heuristic function to guide the search and find the optimal solution in a more efficient manner than UCS. The heuristic should be admissible and consistent, meaning it never overestimates the true distance to the

goal and satisfies the triangle inequality. The best heuristic function for the 8-puzzle problem is the Manhattan distance.

In conclusion, A* search is generally considered the best algorithm for the 8-puzzle problem, as it finds the optimal solution efficiently while ensuring optimality. The average time for A* was approximately 0.0011 which was way better other search algorithms. More results containing the time and space of these algorithms are shown below.

I used Tracemalloc to report the max memory usage for each of the search algorithms. Tracemalloc is a library module that traces every memory block in python. The tracing starts by using the start() during runtime. This library module can also give information about the total size, number, and average size of allocated memory blocks. The output is given in form of (current, peak),i.e, current memory is the memory the code is currently using and peak memory is the maximum space the program used while executing.

☆Bonus part: The best algorithm for solving this problem is A*. I also took a look at IDA* which is Iterative Deepening A*. It is a variant of the A* algorithm that addresses the problem of the high memory usage associated with A*. IDA* uses a depth-first search approach and gradually increases the depth limit while using a heuristic function to guide the search. In general, A* is considered to be a more efficient and flexible algorithm than IDA*. However, IDA* may be preferred in situations where memory usage is a concern, or where the heuristic function satisfies the conditions for monotonicity. Ultimately, the choice between the two algorithms depends on the specific problem and the requirements of the application.

Test cases:

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


DFS Output:  'Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Up', 'Up', 'Right',
'Down', 'Left', 'Down', 'Left', 'Up', 'Right', 'Right',
'Down',
DFS time:  0.634000301361084
DFS Memory usage  (25288, 31448)

IDS Output:  'Up', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down', 'Down',
IDS time:  0.05998539924621582
IDS Memory usage  (1460, 4004)

BFS output:  ['down', 'right', 'up', 'left', 'down', 'right', 'right']
BFS time:  0.04300665855407715
BFS Memory usage  (10439, 1310439)
```

```
UCS output:  ['down', 'right', 'up', 'left', 'down', 'right', 'right']
UCS time:  0.3690016269683838
UCS Memory usage  (10188, 1326156)
AStar
A* Solution is  ['Right', 'Down', 'Left', 'Up', 'Right', 'Down', 'Right']
Astar Time: 0.0020194053649902344
A* Memory usage  (7432, 12464)
---------------------------------------------
Input:  [0, 4, 1, 2, 5, 3, 7, 8, 6]
---------------------------------------------

DFS Output:  'Up', 'Down', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down', 'Left', 'Up',
'Right', 'Up', 'Left', 'Down', 'Right', 'Down', 'Right',
DFS time:  14.503227472305298
DFS Memory usage  (21652, 28740)

IDS Output:  'Up', 'Down', 'Down', 'Right', 'Up', 'Up', 'Left', 'Down', 'Right',
'Down', 'Right',
IDS time:  0.41499948501586914
IDS Memory usage  (5580, 8716)

BFS output:  ['right', 'right', 'down', 'left', 'left', 'up', 'right', 'down', 'right',
'down']
BFS time:  1.5417449474334717
BFS Memory usage  (9884, 31898532)

UCS output:  ['right', 'right', 'down', 'left', 'left', 'up', 'right', 'down', 'right',
'down']
UCS time:  242.26613569259644
UCS Memory usage  (9804, 34947260)
AStar
A* Solution is  ['Right', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Down', 'Right',
'Down']
Astar Time: 0.001997232437133789
A* Memory usage  (2672, 10528)
---------------------------------------------
Input:  [4, 1, 3, 0, 2, 6, 7, 5, 8]
---------------------------------------------
```

```
DFS Output:  'Up', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down', 'Left', 'Up', 'Right',
'Down', 'Down', 'Left', 'Up', 'Up', 'Left', 'Down', 'Right', 'Down', 'Right',
DFS time:  3.4089975357055664
DFS Memory usage  (22428, 27844)

IDS Output:  'Up', 'Left', 'Down', 'Right', 'Right', 'Down',
IDS time:  0.019992351531982422
IDS Memory usage  (896, 3024)

BFS output:  ['up', 'right', 'down', 'down', 'right']
BFS time:  0.005999803543090820
BFS Memory usage  (8852, 112884)

UCS output:  ['up', 'right', 'down', 'down', 'right']
UCS time:  0.009994029998779297
UCS Memory usage  (8908, 113420)
AStar
A* Solution is  ['Up', 'Right', 'Down', 'Down', 'Right']
Astar Time: 0.0020000934600830078
A* Memory usage  (40, 4296)
-------------------------------------------
Input:  [1, 2, 3, 0, 4, 8, 7, 6, 5]
-------------------------------------------

DFS Output:  'Up', 'Right', 'Down', 'Left', 'Down', 'Left', 'Up', 'Right', 'Right', 'Up',
'Left', 'Down', 'Left', 'Down', 'Right', 'Right', 'Up',
'Left', 'Down', 'Right',
DFS time:  2.118001937866211
DFS Memory usage  (11036, 16228)

IDS Output:  'Up', 'Down', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down',
IDS time:  0.06801939010620117
IDS Memory usage  (1460, 4004)

BFS output:  ['right', 'down', 'right', 'up', 'left', 'down', 'right']
BFS time:  0.04899835586547852
BFS Memory usage  (9692, 1726108)
```

```
UCS output:  ['right', 'down', 'right', 'up', 'left', 'down', 'right']
UCS time:  0.6519875526428223
UCS Memory usage  (9740, 1753068)
AStar
A* Solution is  ['Right', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down']
Astar Time: 0.0009999275207519531
A* Memory usage  (1704, 6920)
--------------------------------------------
Input:  [1, 2, 0, 4, 8, 3, 7, 6, 5]
--------------------------------------------

DFS Output:  'Up', 'Right', 'Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Up', 'Left',
'Down', 'Right', 'Up', 'Left', 'Down', 'Right', 'Right',
DFS time:  0.2579994201660156
DFS Memory usage  (20364, 25660)

IDS Output:  'Up', 'Right', 'Right', 'Up', 'Left', 'Down', 'Right',
IDS time:  0.02299976348876953
IDS Memory usage  (1024, 3360)

BFS output:  ['down', 'down', 'left', 'up', 'right', 'down']
BFS time:  0.007996559143066406
BFS Memory usage  (9692, 254324)

UCS output:  ['down', 'down', 'left', 'up', 'right', 'down']
UCS time:  0.02399921417236328
UCS Memory usage  (9692, 256228)
AStar
A* Solution is  ['Down', 'Down', 'Left', 'Up', 'Right', 'Down']
Astar Time: 0.0009999275207519531
A* Memory usage  (48, 4496)
--------------------------------------------
Input:  [1, 0, 2, 4, 6, 3, 7, 5, 8]
--------------------------------------------

DFS Output:  'Up', 'Right', 'Right', 'Up', 'Left', 'Down', 'Right', 'Up', 'Left', 'Down',
'Left', 'Down', 'Right', 'Up', 'Right', 'Up', 'Left', 'Down', 'Right', 'Down',
DFS time:  3.0657737255096436
DFS Memory usage  (11204, 16676)
```

```
IDS Output:  'Up', 'Down', 'Right', 'Up', 'Right', 'Down',
IDS time:  0.021010637283325195
IDS Memory usage  (896, 3024)

BFS output:  ['right', 'down', 'left', 'down', 'right']
BFS time:  0.0059854984283447266
BFS Memory usage  (9452, 227540)

UCS output:  ['right', 'down', 'left', 'down', 'right']
UCS time:  0.02399468421936035
UCS Memory usage  (9452, 229836)
AStar
A* Solution is  ['Right', 'Down', 'Left', 'Down', 'Right']
Astar Time: 0.002005338668823242
A* Memory usage  (40, 4296)
--------------------------------------------
Input:  [0, 1, 2, 4, 5, 3, 7, 8, 6]
--------------------------------------------

DFS Output:  'Up', 'Right', 'Down', 'Right', 'Down', 'Left', 'Up', 'Up', 'Left', 'Down',
'Right', 'Down', 'Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Right',
DFS time:  4.644521951675415
DFS Memory usage  (10908, 17012)

IDS Output:  'Up', 'Down', 'Down', 'Right', 'Right',
IDS time:  0.015004873275756836
IDS Memory usage  (768, 2688)

BFS output:  ['right', 'right', 'down', 'down']
BFS time:  0.0030002593994140625
BFS Memory usage  (6860, 53076)

UCS output:  ['right', 'right', 'down', 'down']
UCS time:  0.004001617431640625
UCS Memory usage  (6420, 52108)
AStar
A* Solution is  ['Right', 'Right', 'Down', 'Down']
Astar Time: 0.0020341873168945312
```

```
A* Memory usage  (32, 3432)
---------------------------------------------
Input:  [1, 2, 3, 0, 4, 5, 7, 8, 6]
---------------------------------------------

DFS Output:  'Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Up', 'Right', 'Up',
'Left', 'Down', 'Left', 'Down', 'Right', 'Up', 'Left', 'Down', 'Right', 'Right',
DFS time:  0.35900378227233887
DFS Memory usage  (9804, 14716)

IDS Output:  'Up', 'Down', 'Down', 'Right',
IDS time:  0.011013507843017578
IDS Memory usage  (576, 2320)

BFS output:  ['right', 'right', 'down']
BFS time:  0.0029995441436767578
BFS Memory usage  (2932, 22644)

UCS output:  ['right', 'right', 'down']
UCS time:  0.0030028820037841797
UCS Memory usage  (1796, 21836)
AStar
A* Solution is  ['Right', 'Right', 'Down']
Astar Time: 0.00099945068359375
A* Memory usage  (24, 3664)
---------------------------------------------
Input:  [1, 2, 3, 4, 0, 5, 7, 8, 6]
---------------------------------------------

DFS Output:  'Up', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down', 'Left', 'Up', 'Right',
'Down',
DFS time:  1.2750003337860107
DFS Memory usage  (7572, 16228)

IDS Output:  'Up', 'Down', 'Right',
IDS time:  0.004998207092285156
IDS Memory usage  (448, 1984)

BFS output:  ['right', 'down']
BFS time:  0.0030066967010498047
BFS Memory usage  (60, 9348)

UCS output:  ['right', 'down']
UCS time:  0.0020225048065185547
UCS Memory usage  (60, 9348)
AStar
A* Solution is  ['Right', 'Down']
Astar Time: 0.0010013580322265625
A* Memory usage  (16, 3224)
--------------------------------------------
Input:  [1, 2, 3, 4, 5, 0, 7, 8, 6]
--------------------------------------------


DFS Output:  'Up', 'Right',
DFS time:  0.0010025501251220703
DFS Memory usage  (320, 1184)

IDS Output:  'Up', 'Right',
IDS time:  0.0009915828704833984
IDS Memory usage  (320, 1312)

BFS output:  ['down']
BFS time:  0.0009827613830566406
BFS Memory usage  (60, 1724)

UCS output:  ['down']
UCS time:  0.0019948482513427734
UCS Memory usage  (60, 1724)
AStar
A* Solution is  ['Down']
Astar Time: 0.0019805431365966797
A* Memory usage  (8, 2392)
--------------------------------------------
Input:  [0, 1, 3, 4, 2, 5, 7, 8, 6]
--------------------------------------------


DFS Output:  'Up', 'Right', 'Down', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down',
'Left', 'Up', 'Up', 'Left', 'Down', 'Right', 'Right', 'Down',
DFS time:  3.786994457244873
DFS Memory usage  (10172, 17012)

IDS Output:  'Up', 'Down', 'Right', 'Down', 'Right',
IDS time:  0.012996912002563477
IDS Memory usage  (768, 2688)

BFS output:  ['right', 'down', 'right', 'down']
BFS time:  0.003000020980834961
BFS Memory usage  (6532, 45396)

UCS output:  ['right', 'down', 'right', 'down']
UCS time:  0.006001949310302734
UCS Memory usage  (6476, 44268)
AStar
A* Solution is  ['Right', 'Down', 'Right', 'Down']
Astar Time: 0.001993894577026367
A* Memory usage  (32, 3864)
--------------------------------------------
Input:  [2, 3, 5, 1, 0, 4, 7, 8, 6]
--------------------------------------------

DFS Output:  'Up', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down', 'Left', 'Up', 'Right',
'Down', 'Left', 'Left', 'Up', 'Up', 'Right', 'Down', 'Down', 'Right',
DFS time:  1.2090015411376953
DFS Memory usage  (10516, 16228)

IDS Output:  'Up', 'Down', 'Left', 'Up', 'Up', 'Right', 'Down', 'Down', 'Right',
IDS time:  0.1399850845336914
IDS Memory usage  (2188, 4908)

BFS output:  ['right', 'up', 'left', 'left', 'down', 'right', 'right', 'down']
BFS time:  0.2089977264404297
BFS Memory usage  (9692, 7306276)

UCS output:  ['right', 'up', 'left', 'left', 'down', 'right', 'right', 'down']
UCS time:  10.310852766036987
UCS Memory usage  (9740, 7786236)
AStar
```

```
A* Solution is  ['Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Right', 'Down']
Astar Time: 0.0020246505737304688
A* Memory usage  (1760, 7168)
--------------------------------------------
Input:  [1, 6, 2, 5, 3, 0, 4, 7, 8]
--------------------------------------------

DFS Output:  'Up', 'Right', 'Up', 'Left', 'Down', 'Right', 'Up', 'Left', 'Down', 'Right',
'Up', 'Left', 'Left', 'Down', 'Right', 'Up', 'Up', 'Right', 'Down', 'Down',
DFS time:  1.4970145225524902
DFS Memory usage  (22428, 27620)

IDS Output:  'Up', 'Up', 'Left', 'Down', 'Right', 'Up', 'Up', 'Right', 'Down', 'Down',
IDS time:  0.19899296760559082
IDS Memory usage  (3716, 6644)

BFS output:  ['left', 'up', 'right', 'down', 'left', 'left', 'down', 'right', 'right']
BFS time:  0.3850102424621582
BFS Memory usage  (9804, 12551052)

UCS output:  ['left', 'up', 'right', 'down', 'left', 'left', 'down', 'right', 'right']
UCS time:  34.230257987976074
UCS Memory usage  (9884, 13658196)
AStar
A* Solution is  ['Left', 'Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Right']
Astar Time: 0.002007007598876953
A* Memory usage  (2344, 9408)
--------------------------------------------
Input:  [1, 8, 2, 0, 4, 3, 7, 6, 5]
--------------------------------------------

DFS Output:  'Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Up', 'Right', 'Up',
'Left', 'Down', 'Right', 'Down', 'Left', 'Up', 'Right',
'Down',
DFS time:  0.40200042724609375
DFS Memory usage  (20884, 26108)

IDS Output:  'Up', 'Down', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Down',
'Right',
IDS time:  0.20400094985961914
IDS Memory usage  (3580, 6476)

BFS output:  ['right', 'up', 'right', 'down', 'down', 'left', 'up', 'right', 'down']
BFS time:  0.4019930362701416
BFS Memory usage  (9804, 12488332)

UCS output:  ['right', 'up', 'right', 'down', 'down', 'left', 'up', 'right', 'down']
UCS time:  36.21971035003662
UCS Memory usage  (9804, 13588924)
AStar
A* Solution is  ['Right', 'Up', 'Right', 'Down', 'Down', 'Left', 'Up', 'Right', 'Down']
Astar Time: 0.0020046234130859375
A* Memory usage  (2136, 8600)
--------------------------------------------
Input:  [2, 5, 3, 4, 1, 6, 0, 7, 8]
--------------------------------------------

DFS Output:  'Up', 'Left', 'Left', 'Down', 'Right', 'Up', 'Left', 'Down', 'Right', 'Right',
'Up', 'Left', 'Down', 'Right', 'Up', 'Left', 'Down', 'Right', 'Down',
DFS time:  2.1899988651275635
DFS Memory usage  (21820, 27172)

IDS Output:  'Up', 'Left', 'Down', 'Left', 'Up', 'Right', 'Right', 'Down', 'Down',
IDS time:  0.09198188781738281
IDS Memory usage  (2220, 4908)

BFS output:  ['up', 'right', 'up', 'left', 'down', 'down', 'right', 'right']
BFS time:  0.06998825073242188
BFS Memory usage  (9692, 2274212)

UCS output:  ['up', 'right', 'up', 'left', 'down', 'down', 'right', 'right']
UCS time:  0.9210002422332764
UCS Memory usage  (9692, 2374500)
AStar
A* Solution is  ['Up', 'Right', 'Up', 'Left', 'Down', 'Down', 'Right', 'Right']
Astar Time: 0.001969575881958008
A* Memory usage  (320, 7592)
--------------------------------------------
```

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

DFS Output:  'Up', 'Left', 'Left', 'Up', 'Right', 'Right', 'Up', 'Left', 'Down', 'Left',
'Down', 'Right', 'Up', 'Up', 'Right', 'Down', 'Down',
DFS time:  0.5939984321594238
DFS Memory usage  (9476, 15164)

IDS Output:  'Up', 'Left', 'Up', 'Right', 'Down',
IDS time:  0.0049974918365478516
IDS Memory usage  (768, 2688)

BFS output:  ['up', 'left', 'down', 'right']
BFS time:  0.00202178955078125
BFS Memory usage  (5652, 31028)

UCS output:  ['up', 'left', 'down', 'right']
UCS time:  0.001986265182495117
UCS Memory usage  (4796, 29932)
AStar
A* Solution is  ['Up', 'Left', 'Down', 'Right']
Astar Time: 0.002019643783569336
A* Memory usage  (32, 3864)
-------------------------------------------
Input:  [1, 6, 2, 5, 7, 3, 0, 4, 8]
-------------------------------------------

DFS Output:  'Up', 'Left', 'Down', 'Right', 'Up', 'Left', 'Down', 'Right', 'Up', 'Left',
'Down', 'Left', 'Down', 'Right', 'Up', 'Up', 'Right', 'Down', 'Down',
DFS time:  4.762003660202026
DFS Memory usage  (10908, 16956)

IDS Output:  'Up', 'Down', 'Left', 'Left', 'Down', 'Right', 'Up', 'Up', 'Right', 'Down',
'Down',
IDS time:  0.2590141296386719
IDS Memory usage  (5580, 8716)

BFS output:  ['right', 'up', 'up', 'right', 'down', 'left', 'left', 'down', 'right', 'right']
BFS time:  0.8749744892120361
BFS Memory usage  (9884, 23970788)

UCS output:  ['right', 'up', 'up', 'right', 'down', 'left', 'left', 'down', 'right', 'right']
UCS time:  131.01605558395386
UCS Memory usage  (9884, 26281396)
AStar
A* Solution is  ['Right', 'Up', 'Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right',
'Right']
Astar Time: 0.002007007598876953
A* Memory usage  (2448, 9904)
-------------------------------------------
Input:  [0, 4, 1, 5, 3, 2, 7, 8, 6]
-------------------------------------------

DFS Output:  'Up', 'Down', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down', 'Left', 'Up',
'Right', 'Up', 'Left', 'Down', 'Down', 'Right', 'Right',
DFS time:  15.538159370422363
DFS Memory usage  (21820, 28908)

IDS Output:  'Up', 'Down', 'Down', 'Right', 'Up', 'Up', 'Left', 'Down', 'Down',
'Right', 'Right',
IDS time:  0.3090941905975342
IDS Memory usage  (5580, 8716)

BFS output:  ['right', 'right', 'down', 'left', 'left', 'up', 'right', 'right', 'down',
'down']
BFS time:  1.373044729232788
BFS Memory usage  (9884, 31906212)

UCS output:  ['right', 'right', 'down', 'left', 'left', 'up', 'right', 'right', 'down',
'down']
UCS time:  247.98274493217468
UCS Memory usage  (9804, 34955732)
AStar
A* Solution is  ['Right', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down',
'Down']
Astar Time: 0.0020031929016113228
A* Memory usage  (2448, 9904)
-------------------------------------------
```

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

DFS Output:  'Up', 'Right', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Right',
DFS time:  0.0029985904693603516
DFS Memory usage  (6512, 8672)

IDS Output:  'Up', 'Right', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Right',
IDS time:  0.07500052452087402
IDS Memory usage  (2204, 4492)

BFS output:  ['down', 'down', 'right', 'up', 'up', 'right', 'down', 'down']
BFS time:  0.06898975372314453
BFS Memory usage  (9692, 2031196)

UCS output:  ['down', 'down', 'right', 'up', 'up', 'right', 'down', 'down']
UCS time:  0.8409996032714844
UCS Memory usage  (9692, 2106292)
AStar
A* Solution is  ['Down', 'Down', 'Right', 'Up', 'Up', 'Right', 'Down', 'Down']
Astar Time: 0.0020000934600830078
A* Memory usage  (64, 5096)
--------------------------------------------
Input:  [1, 2, 3, 0, 4, 6, 7, 5, 8]
--------------------------------------------

DFS Output:  'Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Up', 'Right', 'Up',
'Left', 'Down', 'Down', 'Left', 'Up', 'Right', 'Right',
'Down',
DFS time:  0.4110126495361328
DFS Memory usage  (15348, 20572)

IDS Output:  'Up', 'Down', 'Right', 'Down',
IDS time:  0.007001161575317383
IDS Memory usage  (576, 2320)

BFS output:  ['right', 'down', 'right']
BFS time:  0.001003265380859375
BFS Memory usage  (1492, 18708)
```

```
UCS output:  ['right', 'down', 'right']
UCS time:  0.0020008087158203125
UCS Memory usage  (476, 17900)
AStar
A* Solution is  ['Right', 'Down', 'Right']
Astar Time: 0.002001523971557617
A* Memory usage  (24, 3664)
```

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

DFS: ['Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Up', 'Up', 'Right', 'Down', 'Left', 'Down', 'Left', 'Up', 'Right', 'Right', 'Down']

BFS: ['Right', 'Down', 'Left', 'Up', 'Right', 'Down', 'Right']

IDS: ['Up', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down', 'Down']

A*: ['Right', 'Down', 'Left', 'Up', 'Right', 'Down', 'Right']

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

DFS: ['Up', 'Right', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Right', 'Up', 'Up', 'Left', 'Down', 'Down', 'Right', 'Up', 'Left', 'Down', 'Left', 'Up', 'Up', 'Right', 'Down', 'Right', 'Down']

BFS: ['Right', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Down', 'Right', 'Down']

IDS: ['Up', 'Down', 'Down', 'Right', 'Up', 'Up', 'Left', 'Down', 'Right', 'Down', 'Right']

A*: ['Right', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Down', 'Right', 'Down']

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

DFS: ['Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Down', 'Right', 'Up', 'Left', 'Down', 'Right', 'Up', 'Up', 'Left', 'Down', 'Down', 'Left', 'Up', 'Up', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down', 'Right', 'Down']

BFS: ['Up', 'Right', 'Down', 'Down', 'Right']

IDS: ['Up', 'Left', 'Down', 'Right', 'Right', 'Down']

A*: ['Up', 'Right', 'Down', 'Down', 'Right']

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

DFS: ['Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Down', 'Right', 'Up', 'Up', 'Left', 'Down', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Right']

BFS: ['Right', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down']

IDS: 'Up', 'Down', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down'

A*: ['Right', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down']

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

A*: ['Down', 'Down', 'Left', 'Up', 'Right', 'Down']

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

A*: ['Right', 'Down', 'Left', 'Down', 'Right']

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

A*: ['Right', 'Right', 'Down', 'Down']

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

A*: ['Right', 'Right', 'Down']

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

A*: ['Right', 'Down']

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

A*: ['Down']

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

A*: ['Right', 'Down', 'Right', 'Down']

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

A*: ['Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Right', 'Down']

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

A*: ['Left', 'Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Right']

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

A*: ['Right', 'Up', 'Right', 'Down', 'Down', 'Left', 'Up', 'Right', 'Down']

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

A*: ['Up', 'Right', 'Up', 'Left', 'Down', 'Down', 'Right', 'Right']

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

A*: ['Up', 'Left', 'Down', 'Right']

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

A*: ['Right', 'Up', 'Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Right']

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

A*: ['Right', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down', 'Down']

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

A*: ['Down', 'Down', 'Right', 'Up', 'Up', 'Right', 'Down', 'Down']

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

A*: ['Right', 'Down', 'Right']

Input: [1,3,5,7,2,6,8,0,6]

DFS: ['Up', 'Left', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Right', 'Up', 'Up', 'Left', 'Down', 'Down', 'Right']

BFS: ['Right', 'Up', 'Up', 'Left', 'Down', 'Down', 'Left', 'Up', 'Right', 'Right', 'Down']

IDS: 'Up', 'Down', 'Left', 'Left', 'Up', 'Right', 'Right', 'Up', 'Left', 'Down', 'Down', 'Right'

A*: ['Right', 'Up', 'Up', 'Left', 'Down', 'Down', 'Left', 'Up', 'Right', 'Right', 'Down']

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

DFS: 'Up', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Right', 'Up', 'Left', 'Down', 'Left', 'Down', 'Right', 'Right', 'Up', 'Up', 'Left', 'Down', 'Right', 'Down', 'Left', 'Up', 'Right', 'Up', 'Left', 'Left', 'Down', 'Down', 'Right', 'Right'

BFS: ['Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down', 'Down']

IDS: ['Up', 'Up', 'Right', 'Down', 'Down', 'Left', 'Up', 'Up', 'Left', 'Down', 'Down', 'Right', 'Right']

A*: ['Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down', 'Down']

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

DFS: ['Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Up', 'Right', 'Down', 'Right', 'Down', 'Left', 'Up', 'Up', 'Left', 'Down', 'Right', 'Up', 'Right', 'Down', 'Left', 'Up', 'Left', 'Down', 'Down', 'Right', 'Right']

BFS: ['Right', 'Up', 'Right', 'Down', 'Left', 'Down', 'Left', 'Up', 'Up', 'Right', 'Right', 'Down', 'Down']

IDS: ['Up', 'Down', 'Left', 'Down', 'Right', 'Up', 'Right', 'Up', 'Left', 'Left', 'Down', 'Down', 'Right', 'Right']

A*: ['Right', 'Up', 'Right', 'Down', 'Left', 'Down', 'Left', 'Up', 'Up', 'Right', 'Right', 'Down', 'Down']

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

A*: ['Left', 'Up', 'Up', 'Left', 'Down', 'Right', 'Up', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Down', 'Right', 'Down']

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

A*: ['Down', 'Down', 'Right', 'Up', 'Up', 'Left', 'Left', 'Down', 'Down', 'Right', 'Right', 'Up', 'Left', 'Down', 'Right']

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

A*: ['Right', 'Right', 'Up', 'Up', 'Left', 'Left', 'Down', 'Right', 'Down', 'Right', 'Up', 'Up', 'Left', 'Down', 'Right', 'Down']

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

A*: ['Down', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Up', 'Right', 'Down', 'Down']

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

A∗: ['Up', 'Up', 'Left', 'Down', 'Right', 'Up', 'Right', 'Down', 'Left', 'Up', 'Left', 'Down', 'Down', 'Right', 'Up', 'Right', 'Down']

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

A∗: ['Left', 'Down', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Up', 'Right', 'Down', 'Down', 'Right']

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

A∗: ['Up', 'Up', 'Left', 'Down', 'Right', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down', 'Left', 'Up', 'Left', 'Down', 'Right', 'Right']

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

A∗: ['Up', 'Left', 'Left', 'Down', 'Right', 'Up', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down']

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

A∗: ['Left', 'Left', 'Up', 'Up', 'Right', 'Down', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Up', 'Right', 'Down', 'Down']

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

A∗: ['Left', 'Left', 'Up', 'Right', 'Right', 'Down', 'Left', 'Up', 'Left', 'Down', 'Right', 'Up', 'Right', 'Down']

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

A∗: ['Up', 'Left', 'Up', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down', 'Left', 'Up', 'Left', 'Down', 'Right', 'Right', 'Down']

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

A∗: ['Left', 'Up', 'Left', 'Down', 'Right', 'Up', 'Right', 'Down', 'Left', 'Up', 'Up', 'Right', 'Down', 'Down', 'Left', 'Left', 'Up',

'Right', 'Down', 'Right']

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

DFS: ['Up', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Down', 'Right', 'Up', 'Up', 'Left', 'Down', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down', 'Down']

BFS: ['Left', 'Left', 'Up', 'Right', 'Up', 'Left', 'Down', 'Right', 'Down', 'Right', 'Up', 'Up', 'Left', 'Left', 'Down', 'Down', 'Right', 'Up', 'Up', 'Right', 'Down', 'Down']

IDS: ['Up', 'Left', 'Left', 'Up', 'Right', 'Right', 'Up', 'Left', 'Left', 'Down', 'Down', 'Right', 'Right', 'Up', 'Left', 'Up', 'Left', 'Down', 'Right', 'Up', 'Right', 'Down', 'Down']

A*: ['Left', 'Left', 'Up', 'Right', 'Up', 'Left', 'Down', 'Right', 'Down', 'Right', 'Up', 'Up', 'Left', 'Left', 'Down', 'Down', 'Right', 'Up', 'Up', 'Right', 'Down', 'Down']

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

DFS: ['Up', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down', 'Left', 'Up', 'Right', 'Up', 'Left', 'Down', 'Right', 'Down', 'Left', 'Up', 'Right', 'Up', 'Left', 'Down', 'Right', 'Down', 'Left', 'Up', 'Left', 'Up', 'Right', 'Right', 'Down', 'Down']

BFS: time limit

IDS: time limit

A*: ['Up', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down', 'Left', 'Up', 'Left', 'Down', 'Right', 'Down', 'Right', 'Up', 'Up', 'Left', 'Left', 'Down', 'Right', 'Down', 'Left', 'Up', 'Right', 'Right', 'Down']

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

A*: ['Left', 'Up', 'Right', 'Down', 'Left', 'Up', 'Left', 'Down', 'Right', 'Up', 'Right', 'Down', 'Left', 'Up', 'Left', 'Down', 'Right', 'Up', 'Right', 'Down']

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

A*: ['Right', 'Up', 'Right', 'Down', 'Left', 'Up', 'Left', 'Down', 'Right', 'Up', 'Right', 'Down', 'Left', 'Up', 'Left', 'Down', 'Right', 'Up', 'Right', 'Down']

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

A*: ['Left', 'Up', 'Left', 'Down', 'Down', 'Right', 'Up', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Down', 'Left', 'Up', 'Right', 'Up', 'Left', 'Down', 'Right', 'Down', 'Right']

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

A*: ['Down', 'Left', 'Up', 'Up', 'Left', 'Down', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Up', 'Up', 'Left', 'Down', 'Right', 'Right', 'Down']

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

A*: ['Up', 'Left', 'Down', 'Down', 'Left', 'Up', 'Right', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Down', 'Right', 'Up', 'Left', 'Up', 'Right', 'Down', 'Down']

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

A*: ['Up', 'Left', 'Up', 'Right', 'Down', 'Right', 'Up', 'Left', 'Left', 'Down', 'Down', 'Right', 'Up', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Down', 'Right']

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

A*: ['Left', 'Up', 'Right', 'Up', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Left', 'Down', 'Down', 'Right', 'Up', 'Up', 'Right', 'Down', 'Down']

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

A*: ['Down', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Right', 'Up', 'Left', 'Down', 'Right', 'Down']

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

A*: ['Up', 'Right', 'Down', 'Right', 'Up', 'Left', 'Up', 'Right', 'Down', 'Left', 'Left', 'Down', 'Right', 'Up', 'Left', 'Up', 'Right', 'Right', 'Down', 'Left', 'Down', 'Right']

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

A*: ['Right', 'Down', 'Down', 'Left', 'Left', 'Up', 'Up', 'Right', 'Right', 'Down', 'Down', 'Left', 'Left', 'Up', 'Up', 'Right', 'Right', 'Down', 'Down']

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

A*: ['Down', 'Left', 'Up', 'Left', 'Up', 'Right', 'Down', 'Down', 'Left', 'Up', 'Up', 'Right', 'Down', 'Down', 'Left', 'Up', 'Right',

'Up', 'Right', 'Down', 'Down']

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

A*: ['Up', 'Right', 'Up', 'Left', 'Down', 'Left', 'Down', 'Right', 'Right', 'Up', 'Up', 'Left', 'Left', 'Down', 'Down', 'Right', 'Up', 'Up', 'Right', 'Down', 'Down', 'Left', 'Up', 'Right', 'Down']

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

A*: ['Down', 'Right', 'Right', 'Up', 'Up', 'Left', 'Down', 'Down', 'Left', 'Up', 'Right', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Right', 'Down', 'Left', 'Up', 'Right', 'Up', 'Left', 'Down', 'Down', 'Right']

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

A*: ['Left', 'Up', 'Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Down', 'Right', 'Up', 'Up', 'Left', 'Left', 'Down', 'Right', 'Right', 'Up', 'Left', 'Down', 'Down', 'Right', 'Up', 'Up', 'Left', 'Down', 'Right', 'Down']

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

A*: ['Up', 'Up', 'Right', 'Down', 'Left', 'Up', 'Left', 'Down', 'Right', 'Down', 'Left', 'Up', 'Right', 'Down', 'Right', 'Up', 'Up', 'Left', 'Down', 'Down', 'Right', 'Up', 'Up', 'Left', 'Down', 'Right', 'Down']

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

A*: ['Up', 'Right', 'Down', 'Left', 'Left', 'Up', 'Right', 'Up', 'Left', 'Down', 'Down', 'Right', 'Up', 'Left', 'Up', 'Right', 'Down', 'Right', 'Up', 'Left', 'Left', 'Down', 'Right', 'Right', 'Down']

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

A*: ['Left', 'Left', 'Up', 'Up', 'Right', 'Down', 'Right', 'Down', 'Left', 'Left', 'Up', 'Up', 'Right', 'Down', 'Right', 'Up', 'Left', 'Down', 'Down', 'Right', 'Up', 'Left', 'Down', 'Left', 'Up', 'Up', 'Right', 'Down', 'Right', 'Down']