A\* search: I employed the standard concept of Open and Closed list. The node with smallest f-value from the Open list was transferred to the Closed list and it’s nodes were expanded gradually. When the node to be expanded from the Open list was found to be the goal node, the program was terminated. I applied the Manhattan Heuristic on the problem.

Anytime Weighted A\*: Here I applied the same techniques as the previous one with the exception of the inadmissible heuristic and hence nodes were frequently transferred from the Closed List back to the Open list whenever a path to it with a lower cost was discovered.

IDA\*: Here I simply performed Iterative Depth First Deepening with the node to be selected first being determined by the one with the least evaluation function value. Here the TRICK was to implement a Priority list within a Stack to enable depth first search as well as node selection similar to A\*.

ARA\*: Here I just implemented the algorithm according to the problem. The results were suboptimal and gradually converged to the optimal solution.

Start State = [[3, 6, 9, 4],[5, 2,8, 11], [10, 0, 15, 7],[13, 1, 14, 12]]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | A\* | IDA\* | AWA\*(w=2) | ARA\* |
| Run Time | 8.05612323443 | 1.6174870269 | 392.659 | 21.0691926556 |
| #Nodes inserted into OPEN LIST | 8162 | 3630 | 52199 | 9826 |
| #Nodes expanded | 4044 | 1182 | 46071 | 5791 |
| #Nodes transferred from CLOSED list to OPEN list |  |  | 42 |  |
| #Nodes transferred from CLOSED list to INCONS list |  |  |  | 3 |
| maximum size of INCONS List |  |  |  | 3 |
| maximum size of OPEN+CLOSED+INCONS List |  |  |  | 8135 |
| Max Size of OPEN & CLOSED List | 8069 | 1208 | 46501 |  |
| Length of Optimal Path | 29 | 29 | 29 | 29 |
| Error Bound |  |  | N/A |  |

SOLUTION:

'[[3, 6, 9, 4], [5, 2, 8, 11], [10, 0, 15, 7], [13, 1, 14, 12]]',

'[[3, 6, 9, 4], [5, 2, 8, 11], [10, 1, 15, 7], [13, 0, 14, 12]]',

'[[3, 6, 9, 4], [5, 2, 8, 11], [10, 1, 15, 7], [13, 14, 0, 12]]',

'[[3, 6, 9, 4], [5, 2, 8, 11], [10, 1, 0, 7], [13, 14, 15, 12]]',

'[[3, 6, 9, 4], [5, 2, 8, 11], [10, 1, 7, 0], [13, 14, 15, 12]]',

'[[3, 6, 9, 4], [5, 2, 8, 0], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[3, 6, 9, 4], [5, 2, 0, 8], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[3, 6, 0, 4], [5, 2, 9, 8], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[3, 0, 6, 4], [5, 2, 9, 8], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[0, 3, 6, 4], [5, 2, 9, 8], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[5, 3, 6, 4], [0, 2, 9, 8], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[5, 3, 6, 4], [2, 0, 9, 8], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[5, 3, 6, 4], [2, 9, 0, 8], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[5, 3, 0, 4], [2, 9, 6, 8], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[5, 0, 3, 4], [2, 9, 6, 8], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[0, 5, 3, 4], [2, 9, 6, 8], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[2, 5, 3, 4], [0, 9, 6, 8], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[2, 5, 3, 4], [9, 0, 6, 8], [10, 1, 7, 11], [13, 14, 15, 12]]',

'[[2, 5, 3, 4], [9, 1, 6, 8], [10, 0, 7, 11], [13, 14, 15, 12]]',

'[[2, 5, 3, 4], [9, 1, 6, 8], [0, 10, 7, 11], [13, 14, 15, 12]]',

'[[2, 5, 3, 4], [0, 1, 6, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[2, 5, 3, 4], [1, 0, 6, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[2, 0, 3, 4], [1, 5, 6, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[0, 2, 3, 4], [1, 5, 6, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [0, 5, 6, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 0, 6, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 0, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 0, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 0], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 0]]'

Start State : [[9, 5, 7, 4],[1, 0, 3, 8], [13, 10, 2, 12],[14, 6, 11, 15]]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | A\* | IDA\* | AWA\*(w=1.01,1.2) | ARA\* |
| Run Time | 0.362711 | 0.734927 | 2.940708467 | 1.093 |
| #Nodes inserted into OPEN LIST | 905 | 2296 | 3719 | 1825 |
| #Nodes expanded | 438 | 733 | 1392 | 869 |
| #Nodes transferred from CLOSED list to OPEN list |  |  | 0 |  |
| #Nodes transferred from CLOSED list to INCONS list |  |  |  | 1 |
| maximum size of INCONS List |  |  |  | 1 |
| maximum size of OPEN+CLOSED+INCONS List |  |  |  | 1783 |
| Max Size of OPEN & CLOSED List | 901 | 752 | 3720 |  |
| Length of Optimal Path | 22 | 22 | 22 | 22 |
| Error Bound |  |  | N/A |  |

Solution:

'[[9, 5, 7, 4], [1, 0, 3, 8], [13, 10, 2, 12], [14, 6, 11, 15]]',

'[[9, 5, 7, 4], [0, 1, 3, 8], [13, 10, 2, 12], [14, 6, 11, 15]]',

'[[0, 5, 7, 4], [9, 1, 3, 8], [13, 10, 2, 12], [14, 6, 11, 15]]',

'[[5, 0, 7, 4], [9, 1, 3, 8], [13, 10, 2, 12], [14, 6, 11, 15]]',

'[[5, 1, 7, 4], [9, 0, 3, 8], [13, 10, 2, 12], [14, 6, 11, 15]]',

'[[5, 1, 7, 4], [9, 3, 0, 8], [13, 10, 2, 12], [14, 6, 11, 15]]',

'[[5, 1, 7, 4], [9, 3, 2, 8], [13, 10, 0, 12], [14, 6, 11, 15]]',

'[[5, 1, 7, 4], [9, 3, 2, 8], [13, 0, 10, 12], [14, 6, 11, 15]]',

'[[5, 1, 7, 4], [9, 3, 2, 8], [13, 6, 10, 12], [14, 0, 11, 15]]',

'[[5, 1, 7, 4], [9, 3, 2, 8], [13, 6, 10, 12], [0, 14, 11, 15]]',

'[[5, 1, 7, 4], [9, 3, 2, 8], [0, 6, 10, 12], [13, 14, 11, 15]]',

'[[5, 1, 7, 4], [0, 3, 2, 8], [9, 6, 10, 12], [13, 14, 11, 15]]',

'[[0, 1, 7, 4], [5, 3, 2, 8], [9, 6, 10, 12], [13, 14, 11, 15]]',

'[[1, 0, 7, 4], [5, 3, 2, 8], [9, 6, 10, 12], [13, 14, 11, 15]]',

'[[1, 3, 7, 4], [5, 0, 2, 8], [9, 6, 10, 12], [13, 14, 11, 15]]',

'[[1, 3, 7, 4], [5, 2, 0, 8], [9, 6, 10, 12], [13, 14, 11, 15]]',

'[[1, 3, 0, 4], [5, 2, 7, 8], [9, 6, 10, 12], [13, 14, 11, 15]]',

'[[1, 0, 3, 4], [5, 2, 7, 8], [9, 6, 10, 12], [13, 14, 11, 15]]',

'[[1, 2, 3, 4], [5, 0, 7, 8], [9, 6, 10, 12], [13, 14, 11, 15]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 0, 10, 12], [13, 14, 11, 15]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 0, 12], [13, 14, 11, 15]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 0, 15]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 0]]'

Start State: [[5, 3, 0, 4], [7, 2, 6, 8], [1, 9, 10, 11], [13, 14, 15, 12]]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | A\* | IDA\* | AWA\* | ARA\* |
| Run Time | 3.268297 | 0.33076858 |  | 2.208974492 |
| #Nodes inserted into OPEN LIST | 4739 | 866 |  | 2796 |
| #Nodes expanded | 2304 | 275 |  | 1475 |
| #Nodes transferred from CLOSED list to OPEN list |  |  |  |  |
| #Nodes transferred from CLOSED list to INCONS list |  |  |  | 3 |
| maximum size of INCONS List |  |  |  | 3 |
| maximum size of OPEN+CLOSED+INCONS List |  |  |  | 1849 |
| Max Size of OPEN & CLOSED List | 4702 | 281 |  |  |
| Length of Optimal Path | 22 | 22 |  | 22 |
| Error Bound |  |  |  |  |

Solution:

'[[5, 3, 0, 4], [7, 2, 6, 8], [1, 9, 10, 11], [13, 14, 15, 12]]',

'[[5, 0, 3, 4], [7, 2, 6, 8], [1, 9, 10, 11], [13, 14, 15, 12]]',

'[[5, 2, 3, 4], [7, 0, 6, 8], [1, 9, 10, 11], [13, 14, 15, 12]]',

'[[5, 2, 3, 4], [0, 7, 6, 8], [1, 9, 10, 11], [13, 14, 15, 12]]',

'[[0, 2, 3, 4], [5, 7, 6, 8], [1, 9, 10, 11], [13, 14, 15, 12]]',

'[[2, 0, 3, 4], [5, 7, 6, 8], [1, 9, 10, 11], [13, 14, 15, 12]]',

'[[2, 3, 0, 4], [5, 7, 6, 8], [1, 9, 10, 11], [13, 14, 15, 12]]',

'[[2, 3, 6, 4], [5, 7, 0, 8], [1, 9, 10, 11], [13, 14, 15, 12]]',

'[[2, 3, 6, 4], [5, 0, 7, 8], [1, 9, 10, 11], [13, 14, 15, 12]]',

'[[2, 3, 6, 4], [0, 5, 7, 8], [1, 9, 10, 11], [13, 14, 15, 12]]',

'[[2, 3, 6, 4], [1, 5, 7, 8], [0, 9, 10, 11], [13, 14, 15, 12]]',

'[[2, 3, 6, 4], [1, 5, 7, 8], [9, 0, 10, 11], [13, 14, 15, 12]]',

'[[2, 3, 6, 4], [1, 5, 7, 8], [9, 10, 0, 11], [13, 14, 15, 12]]',

'[[2, 3, 6, 4], [1, 5, 0, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[2, 3, 0, 4], [1, 5, 6, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[2, 0, 3, 4], [1, 5, 6, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[0, 2, 3, 4], [1, 5, 6, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [0, 5, 6, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 0, 6, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 0, 8], [9, 10, 7, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 0, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 0], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 0]]'

Start State: [[6, 2, 4, 8], [15, 9, 1, 0], [7, 5, 11, 3], [14, 13, 10, 12]]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | A\* | IDA\* | AWA\* | ARA\* |
| Run Time | 16.556741 | 5.22757 |  | 146.506 |
| #Nodes inserted into OPEN LIST | 20620 | 17748 |  | 37489 |
| #Nodes expanded | 10384 | 5791 |  | 19791 |
| #Nodes transferred from CLOSED list to OPEN list |  |  |  |  |
| #Nodes transferred from CLOSED list to INCONS list |  |  |  | 153 |
| maximum size of INCONS List |  |  |  | 132 |
| maximum size of OPEN+CLOSED+INCONS List |  |  |  | 27519 |
| Max Size of OPEN & CLOSED List | 20431 | 5817 |  |  |
| Length of Optimal Path | 34 | 36 |  | 34 |
| Error Bound |  |  |  |  |

Solution: A\* and ARA\*

[[6, 2, 4, 8], [15, 9, 1, 0], [7, 5, 11, 3], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [15, 9, 1, 3], [7, 5, 11, 0], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [15, 9, 1, 3], [7, 5, 0, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [15, 9, 1, 3], [7, 0, 5, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [15, 9, 1, 3], [0, 7, 5, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [0, 9, 1, 3], [15, 7, 5, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [9, 0, 1, 3], [15, 7, 5, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [9, 1, 0, 3], [15, 7, 5, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [9, 1, 5, 3], [15, 7, 0, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [9, 1, 5, 3], [15, 0, 7, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [9, 1, 5, 3], [0, 15, 7, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [9, 1, 5, 3], [14, 15, 7, 11], [0, 13, 10, 12]]',

'[[6, 2, 4, 8], [9, 1, 5, 3], [14, 15, 7, 11], [13, 0, 10, 12]]',

'[[6, 2, 4, 8], [9, 1, 5, 3], [14, 0, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [9, 1, 5, 3], [0, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [0, 1, 5, 3], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [1, 0, 5, 3], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [1, 5, 0, 3], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [1, 5, 3, 0], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 0], [1, 5, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 0, 4], [1, 5, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 0, 2, 4], [1, 5, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[0, 6, 2, 4], [1, 5, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[1, 6, 2, 4], [0, 5, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[1, 6, 2, 4], [5, 0, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[1, 0, 2, 4], [5, 6, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[1, 2, 0, 4], [5, 6, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[1, 2, 3, 4], [5, 6, 0, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 14, 0, 11], [13, 15, 10, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 14, 10, 11], [13, 15, 0, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 14, 10, 11], [13, 0, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 0, 10, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 0, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 0], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 0]]

Solution in IDA\*:

[[6, 2, 4, 8], [15, 9, 1, 0], [7, 5, 11, 3], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [15, 9, 1, 3], [7, 5, 11, 0], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [15, 9, 1, 3], [7, 5, 0, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [15, 9, 1, 3], [7, 0, 5, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [15, 9, 1, 3], [0, 7, 5, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [0, 9, 1, 3], [15, 7, 5, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [9, 0, 1, 3], [15, 7, 5, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [9, 7, 1, 3], [15, 0, 5, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [9, 7, 1, 3], [0, 15, 5, 11], [14, 13, 10, 12]]',

'[[6, 2, 4, 8], [9, 7, 1, 3], [14, 15, 5, 11], [0, 13, 10, 12]]',

'[[6, 2, 4, 8], [9, 7, 1, 3], [14, 15, 5, 11], [13, 0, 10, 12]]',

'[[6, 2, 4, 8], [9, 7, 1, 3], [14, 0, 5, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [9, 0, 1, 3], [14, 7, 5, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [9, 1, 0, 3], [14, 7, 5, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [9, 1, 5, 3], [14, 7, 0, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [9, 1, 5, 3], [14, 0, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [9, 1, 5, 3], [0, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [0, 1, 5, 3], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [1, 0, 5, 3], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [1, 5, 0, 3], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 8], [1, 5, 3, 0], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 4, 0], [1, 5, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 2, 0, 4], [1, 5, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[6, 0, 2, 4], [1, 5, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[0, 6, 2, 4], [1, 5, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[1, 6, 2, 4], [0, 5, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[1, 6, 2, 4], [5, 0, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[1, 0, 2, 4], [5, 6, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[1, 2, 0, 4], [5, 6, 3, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[1, 2, 3, 4], [5, 6, 0, 8], [9, 14, 7, 11], [13, 15, 10, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 14, 0, 11], [13, 15, 10, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 14, 10, 11], [13, 15, 0, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 14, 10, 11], [13, 0, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 0, 10, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 0, 11], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 0], [13, 14, 15, 12]]',

'[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 0]]

4.

Comparative analysis:

The IDA\* algorithm seems to be working fastest to provide a solution and with lesser space requirements. However we have to bear with repeated state evaluation on different iterations. On the other hand AWA\* fails to converge to an optimal solution for majority of the tested configurations. ARA\* also provides a fast optimal solution and is better off wrt to IDA\* since we do not move nodes from CLOSESD LIST to OPEN LIST but rather to the INCONS list which is the reason behind being able to gradually change the heuristic function from a inadmissible to admissible and at the same time to transform the function to monotonous and give an optimal result from a suboptimal.

The results provided above are with respect to Manhattan heuristic. It provided better results than simply considering the number of tiles misplaced since they are always not a true cost of the exact heuristic as way longer sequence of steps need to be taken to find the optimal path. But on the other hand, the Manhattan heuristic is well informed.