Burak Aybi

20712

1. Initial state: S

Final state: G

Successors:

If block is up: (x, y-2, x, y-1), x, y+1, x, y+2), (x-2, y, x-1, y), (x+1, y, x+2, y)

If block is vertical: (x, y-1), (x, y + 1), x - 1, y, x - 1, y), (x + 1, y, x + 1, y)

If block is horizontal: (x, y - 1,x, y - 1), (x, y + 1,x, y + 1),( x - 1, y), (x + 1, y)

1. With AStar.py .checking all neighbors to achieve the goal(g). It uses x and y values to find shortest way.
2. Fist board: AStar.py 0.02 second and path: [[(3, 2), (3, 3)], [(4, 2), (4, 3)], [(5, 2), (5, 3)], [(6, 2), (6, 3)], [(7, 2), (7, 3)], [(7, 4)]]

UCS.py 0.036 second and goal point (4,7) and path: [(3, 2), (3, 3)], [(4, 2), (4, 3)], [(5, 2), (5, 3)], [(6, 2), (6, 3)], [(7, 2), (7, 3)], [(7, 4)]

Second board: AStar.py 0.041 second and path: [(0, 1), (0, 2)], [(1, 1), (1, 2)], [(1, 3)], [(2, 3), (3, 3)], [(4, 3)], [(5, 3), (6, 3)], [(5, 4), (6, 4)], [(7, 4)]

UCS.py 0.082 second and path: [(0, 1), (0, 2)], [(1, 1), (1, 2)], [(1, 3)], [(2, 3), (3, 3)], [(4, 3)], [(5, 3), (6, 3)], [(5, 4), (6, 4)], [(7, 4)]

Third board: AStar.py 0.083 second and path: [(3, 2), (4, 2)], [(5, 2)], [(5, 3), (5, 4)], [(5, 5)], [(5, 6), (5, 7)], [(4, 6), (4, 7)], [(4, 8)], [(2, 8), (3, 8)], [(2, 7), (3, 7)], [(4, 7)]

UCS.py 0.243 second and path: [(3, 2), (4, 2)], [(5, 2)], [(5, 3), (5, 4)], [(5, 5)], [(5, 6), (5, 7)], [(4, 6), (4, 7)], [(4, 8)], [(2, 8), (3, 8)], [(2, 7), (3, 7)], [(4, 7)]

UCS.py has more time to complete time. It is surprising because algorithms are similar but there are great difference