

## ***Read Me***

To generate random five solvable initial states:  
(random-case)

The output is as following:

((0 8 2 1 3 6 7 4 5) (0 4 2 5 6 8 3 1 7) (3 5 1 4 2 8 6 7 0) (6 1 5 2 4 8 7 3 0) (3 5 1 7 0  
8 6 4 2))

To change heuristics function:  
(8-puzzle 'start-state '#heuristic)

My three heuristic functions is as following( onvenience, I combine the heuristic function with cost value, they are actually evaluation function).

misplaced-----misplace tiles heuristic

manhattan-----manhattan distance heuristic

extracredit-----bonus: improve the manhattan heuristic function, modify the original manhattan heuristic function to solve the linear conflict. Given two tiles in their goal row. but reversed in position. additional vertical moves can be added to Manhattan distance.

## ***My bonus Heuristics Function***

My Heuristic Function is based on Manhattan Distance. The basic idea is considering more specifically how the tile moves.

For example the state is :

0 2 1

3 4 5

6 7 8

the Manhattan distance is 2 but in fact even if there are all blanks except where Tile 1 and Tile 2 stay, it still need 4 moves to make 1,2 into the correct order. So my idea is add Manhattan distance with pair of tiles that are in the same row/column and are reversed with each other. The value of this heuristic function is always not smaller than Manhattan distance and would not be larger than the real steps. Thus I think it is consistent and dominate Manhattan distance.

## ***Test Report***

### ***Random Test 1***

CL-USER> (8-puzzle '(0 8 2 1 3 6 7 4 5) #'misplaced)

((("down" "right" "up" "left" "down" "down" "right" "up" "right" "down" "left" "up" "left" "down" "right" "up" "left" "up") 18 2682)

CL-USER> (8-puzzle '(0 8 2 1 3 6 7 4 5) #'manhattan)  
(("down" "right" "up" "left" "down" "down" "right" "up" "right" "down" "left" "left"  
"up" "right" "down" "left" "up" "up") 18 482)

CL-USER> (8-puzzle '(0 8 2 1 3 6 7 4 5) #'extracredit)  
(("down" "right" "up" "left" "down" "down" "right" "up" "right" "down" "left" "left"  
"up" "right" "down" "left" "up" "up") 18 482)

### ***Random Test 2***

CL-USER> (8-puzzle '(0 4 2 5 6 8 3 1 7) #'misplaced)  
(("down" "down" "right" "up" "left" "up" "right" "down" "left" "down" "right" "right"  
"up" "left" "left" "up" "right" "down" "left" "up") 20 6217)

CL-USER> (8-puzzle '(0 4 2 5 6 8 3 1 7) #'manhattan)  
(("down" "down" "right" "up" "left" "up" "right" "down" "left" "down" "right" "right"  
"up" "left" "left" "up" "right" "down" "left" "up") 20 1142)

CL-USER> (8-puzzle '(0 4 2 5 6 8 3 1 7) #'extracredit)  
(("down" "down" "right" "up" "left" "up" "right" "down" "left" "down" "right" "right"  
"up" "left" "up" "left" "down" "right" "up" "left") 20 1429)

### ***Random Test 3***

CL-USER> (8-puzzle '(3 5 1 4 2 8 6 7 0) #'misplaced)  
(("up" "left" "up" "right" "down" "left" "left" "up") 8 34)

CL-USER> (8-puzzle '(3 5 1 4 2 8 6 7 0) #'manhattan)  
(("up" "left" "up" "right" "down" "left" "left" "up") 8 29)

CL-USER> (8-puzzle '(3 5 1 4 2 8 6 7 0) #'extracredit)  
(("up" "left" "up" "right" "down" "left" "left" "up") 8 24)

### ***Random Test 4***

CL-USER> (8-puzzle '(6 1 5 2 4 8 7 3 0) #'misplaced)  
(("up" "left" "left" "up" "right" "down" "down" "left" "up" "right" "right" "up" "left"  
"left") 14 459)

CL-USER> (8-puzzle '(6 1 5 2 4 8 7 3 0) #'manhattan)  
(("up" "left" "left" "up" "right" "down" "down" "left" "up" "right" "right" "up" "left"  
"left") 14 109)

CL-USER> (8-puzzle '(6 1 5 2 4 8 7 3 0) #'extracredit)  
(("up" "left" "left" "up" "right" "down" "down" "left" "up" "right" "right" "up" "left"  
"left") 14 118)

### ***Random Test 5***

CL-USER> (8-puzzle '(3 5 1 7 0 8 6 4 2) #'misplaced)

((("right" "down" "left" "up" "up" "right" "down" "left" "left" "up" "right" "right"  
"down" "down" "left" "up" "right" "up" "left" "left") 20 12195)

CL-USER> (8-puzzle '(3 5 1 7 0 8 6 4 2) #'manhattan)

((("right" "down" "left" "up" "up" "right" "down" "left" "left" "up" "right" "right"  
"down" "down" "left" "up" "right" "up" "left" "left") 20 1331)

CL-USER> (8-puzzle '(3 5 1 7 0 8 6 4 2) #'extracredit)

((("right" "down" "left" "up" "up" "right" "down" "left" "left" "up" "right" "right"  
"down" "down" "left" "up" "right" "up" "left" "left") 20 1818)

### ***Conclusion & Analysis***

	Test 1	Test 2	Test 3	Test 4	Test 5
Manhattan Distance	482	1142	29	109	1331
extracredit	482	1429	24	118	1818
Misplaced Tiles	2682	6217	34	459	12159

It is obvious that the number of expanded nodes based on Misplaced is much larger than that based on the other two method.

However the performance of Manhattan Distance plus function is not much better than the original Manhattan Distance method introduced in lectures. What's worse, in some case, Manhattan Distance plus performs worse than the original Manhattan Distance. I think it is because the difference between these two methods is really tiny.