

Apply A* Algorithm

Misplaced
Tiles.

Q	8	3
1	6	4
7	5	.

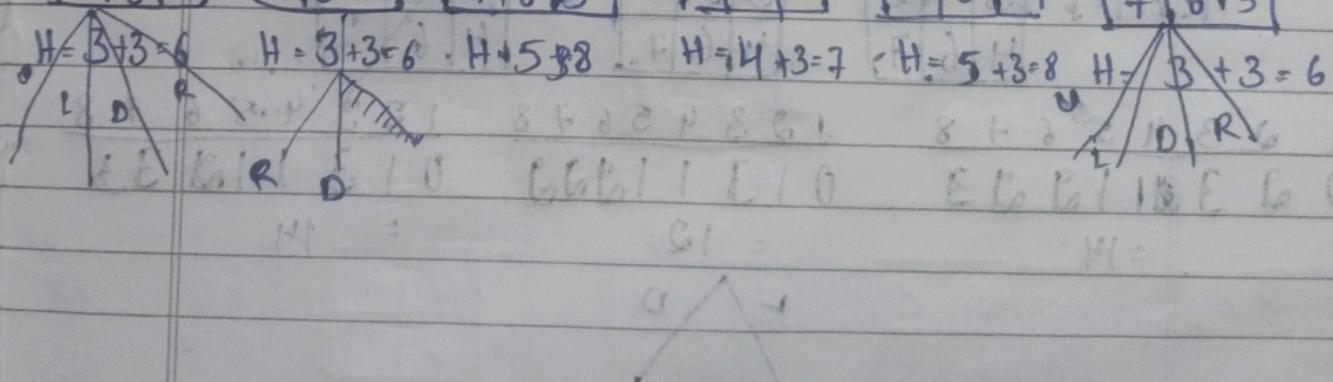
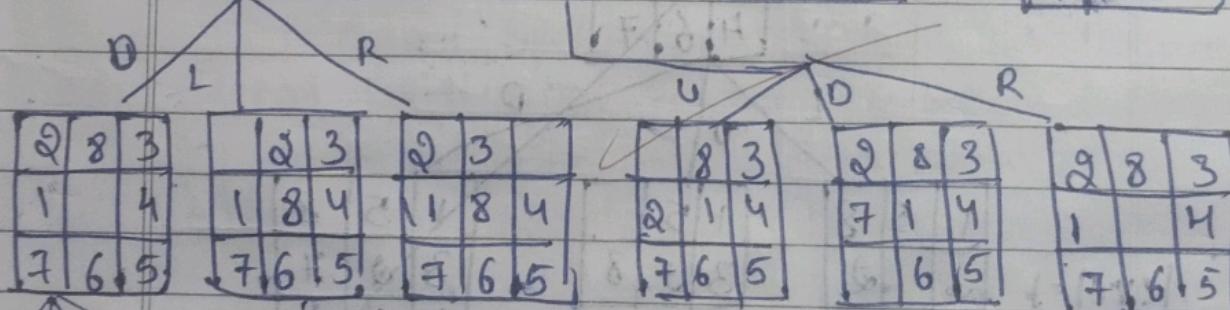
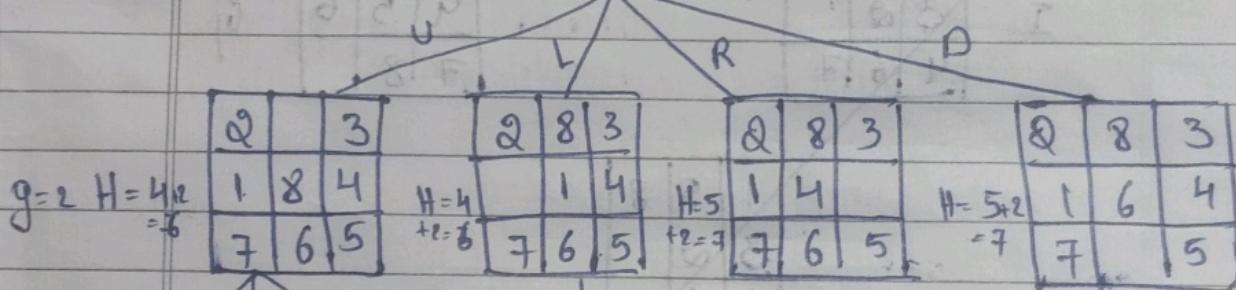
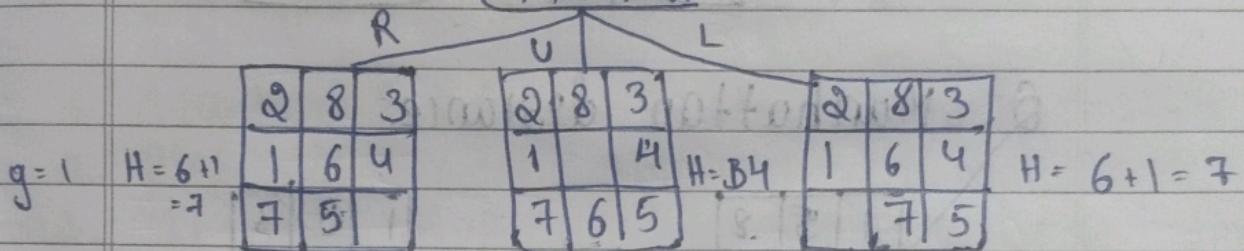
manhattan
distance

1	2	3
8	.	4
7	6	5

$$f(n) = g(n) + h(n)$$

2	8	3
1	6	4
7	.	5

$$g = 0$$



$g=4$ $H=6$

2	8	3
1	4	
7	6	5

 $H=6$

1	8	3
1	8	4
7	6	5

2	8	3
1	4	
7	6	5

 $H=8$ U D $H=8$ R $H=9$ $H=9$ $H=6$ $H=8$ L U R

2 8 3	2 8 3	2 8 3	2 8 3	2 8 3	2 8 3	2 8 3	2 8 3	2 8 3
1 8 4	1 6 4	1 4	1 4	1 8 4	8 4	1 4	1 4	1 6 4
7 6 5	7 5	7 6 5	7 6 5	7 6 5	7 6 5	7 6 5	7 6 5	7 6 5

(a) + (a)

 D R $g=5$

1	8	3
7	8	4
6	5	7

1	8	3
8		4
7	6	5

Q. Manhattan distance.

1	5	8
3	2	
4	6	7

1	9	3
4	5	6
7	8	

1	5	8
3	2	
4	6	7

1	5	8
3	2	
4	6	7

1	5	8
3	2	7
4	6	

1 2 3 4 5 6 7 8

0 2 3 1 1 2 2 3

 $= 14$

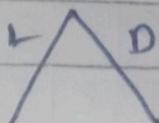
1 2 3 4 5 6 7 8

0 1 3 1 1 2 2 2

 $= 12$

1 2 3 4 5 6 7 8

0 1 3 1 1 2 2 3

 $= 14$ 

1	5	
3	2	8
4	6	7

L D

1	5	
3	2	8
4	6	7

level = 2

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

= 13

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

= 13

Algo for Misplaced tiles

1. Start \rightarrow put initial state in OPEN ($f = g + h$)
2. Pick state with smallest f
3. If goal stop
4. Expand neighbors
5. for each neighbor
 - g : = parent.g + 1
 - h : = misplaced tiles + down
 - f : = $g + h$
 - add / update in Open
6. Repeat until goal found or open entry empty.

Output for Misplaced tiles.

0	8	3
1	6	4
7	0	5

2	3	6
5	4	7
8	9	1

8	7	6	5	4	3	2	1	0
3	0	1	1	F	6	5	4	3
0	0	1	1	N	F	6	5	4

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

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

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

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

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

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

Srushti N
1BM24CS424

2	8	3
1	0	4
7	6	5

3	1	2
8	6	7
5	8	9

2	8	3
1	8	4
7	6	5

8	7	1
8	8	1
8	6	8

initial

F	8	H
F	8	H

8	6	2
0	8	3
7	6	5

8	5	2
8	3	4
8	6	6

El

1	0	3
0	8	4
7	6	5

(cost 1000) goal 000

1	0	3
8	0	4
7	6	5

goal 1000 IT

add 1000 state 1000

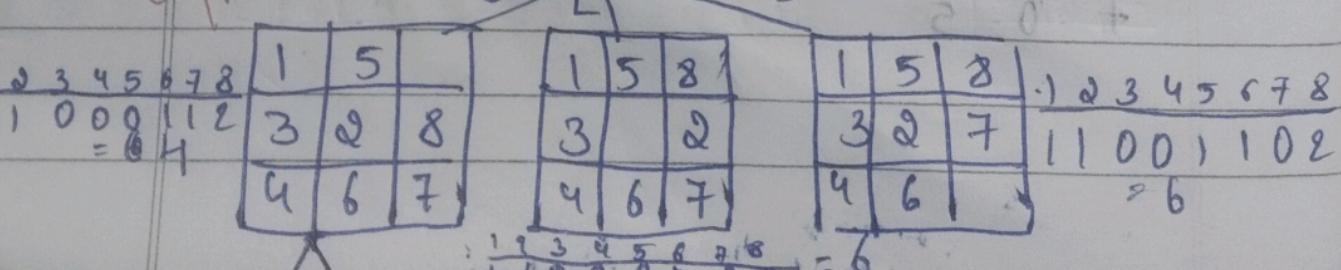
add 1000 state 1000

Manhattan distance

2	8	38
3	6	4
7	0	45

1	2	3
8	5	4
7	8	65

2	8	38
3	6	4
7	0	57



d	91
Q 3	Q 8 3
1 8 4	1 6 4
7 6 5	7 5

1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
1 1 0 0 0 0 0 1 1 1 0 0 0 1 0 2 1 1 0 1 0 0 0 2 2 1 0 0 0 0 0 2
= 3 = 5 = 5 = 5

d	a	L
Q 3	2 3	9 8 3
1 8 4	1 8 4	1 4
7 6 5	7 6 5	7 6 5

1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8 1 2 3 4 5 6 7 8
1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 2
= 2 = 4 = 4

R	D
Q 3	1 2 3
1 8 4	8 4
7 6 5	7 6 5

1 2 3 4 5 6 7 8 = 1
6 0 0 0 0 0 0 1

R	V	R	D
1 2 3 4 5 6 7 8	= 3	1 2 3 4 5 6 7 8	
1 1 0 0 0 0 0 1		6 0 0 0 0 0 0 1	= 1
8	1 2 3	1 2 3 4 5 6 7 8	
0	8 4	6 0 0 0 0 0 0 6	
7 6 5	7 6 5	1 2 3 4 5 6 7 8	
Q 3	V	1 2 3	
1 8 4		7 8 4	
7 6 5		6 5	

Algorithm for Manhattan distance

1. start with the current puzzle state.
2. Create an empty list to keep track of file matches.
3. For each file from 1 to 8:
 - if the file is in cur pos, add 0 to the list.
 - otherwise, add 1 to the list.
4. Check the list:
 - if the list contains all 0's the puzzle is solved → stop.
 - otherwise, make a move to get closer to the goal.
5. Repeat steps 2 to 4 until the puzzle is solved

Output:

Solution path:

2	8	3
1	6	4
7	0	5

↓

2	8	3
1	0	4
7	6	5

↓

2	0	3
1	8	4
7	6	5

↓

0	2	3
1	8	4
7	6	5

↓

1	2	3
0	8	4
7	6	5

↓

1	2	3
8	0	4
7	6	5

↓

Solution path:

(2, 8, 3)

(1, 6, 4)

(7, 0, 5)

(2, 8, 3)

(1, 0, 4)

(7, 6, 5)

(2, 0, 3)

(1, 8, 4)

(7, 6, 5)

(0, 2, 3)

(1, 8, 4)

(7, 6, 5)

(1, 2, 3)

(0, 8, 4)

(7, 6, 5)

(1, 2, 3)

(8, 0, 4)

(7, 6, 5)

**Srushti N
1BM24CS424**