

# ★ Solving 8-puzzle problem using A\* and Implementing Iterative Deeping Search Algorithm

Step 1 : initialize the Problem  
 'define the initial state in a puzzle of (3x3 grid)  
 'define Goal state in a puzzle of (3x3 grid)  
 'determining the empty tile → '0'

Initial State

1	2	3
8	0	4
7	6	5

Final State

2	8	1
0	4	3
7	6	5

Now, getting the lowest value using priority queue  
 Calculating each tile from initial state to Final State.

	<u>V</u>	<u>H</u>	Total Distance	Priority Queue
1	0	2	2	2
2	0	1	1	1
3	1	0	1	1
8	1	1	2	2
4	0	1	1	1
7	0	0	0	0
6	0	0	0	0
5	0	0	0	0

Apply this to priority queue.



So,

considering the lowest cost =  $g(n)$   
~~an~~ estimated cost =  $h(n)$

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

$$\text{lowest distance} = |i - \text{goal } i| + |j - \text{goal } j|$$

fetching neighbour state moves

$(0, 1)$   $(1, 0)$   $(-1, 0)$   $(0, -1)$

Initial State

Step 1

1	2	3
8	0	4
7	6	5

Step 2

1	0	3
8	2	4
7	6	5

Step 3

0	1	3
8	2	4
7	6	5

Step 4

8	1	3
0	2	4
7	6	5



Step 5:

8	1	3
2	0	4
7	6	5

Step 6:

8	1	3
2	4	0
7	6	5

Step 7:

8	1	0
2	4	3
7	6	5

Step 8:

8	0	1
2	4	3
7	6	5

Step 9:

0	8	1
2	4	3
7	6	5

Step 10:

2	8	1
0	4	3
7	6	5

{Goal State}

//