

15/10/24

Week - 3

PAGE NO: 11

DATE:

Implement A* Search Algorithm.

Time Complexity (A* 8 puzzle Problem) $= O(b^d)$

I. Using Misplaced Tiles Heuristic

1. Initialize

- Define the initial state and the goal state
- Create an open list to keep track of nodes to be evaluated, initialized with the starting state
- Create a set to track visited states.

2. While Open list is Not Empty.

- Pop the state with the lowest $[f(n) = g(n) + h(n)]$ from the open list.
- If this state is the goal state, print the path and return the number of moves.

Generate Neighbors

- For each neighbor generated by moving the blank tile
 - Calculate $g(n)$ (depth of the current state + 1).
 - Calculate $h(n)$ (number of misplaced tiles).
 - Compute $f(n) = g(n) + h(n)$.
 - If the neighbor has not been visited, add it to open list and mark it as visited.

3. Sort the open list based on $f(n)$ for the next iteration.

4. If the open list is empty and no solution is found, return -1.

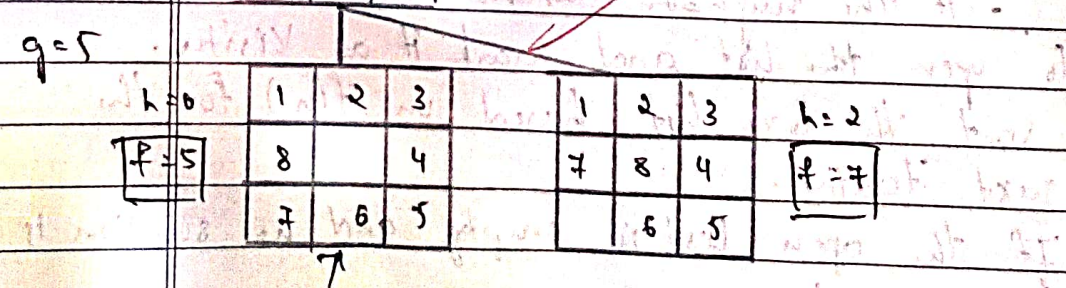
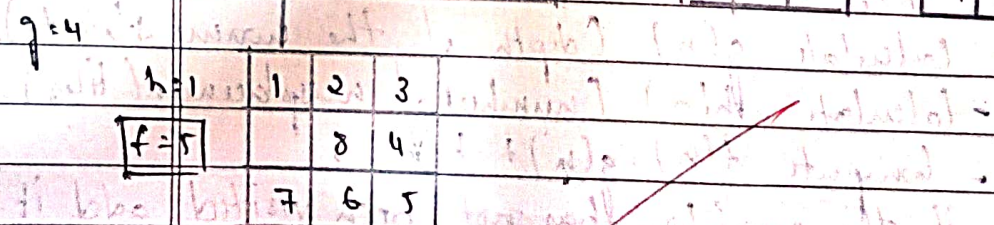
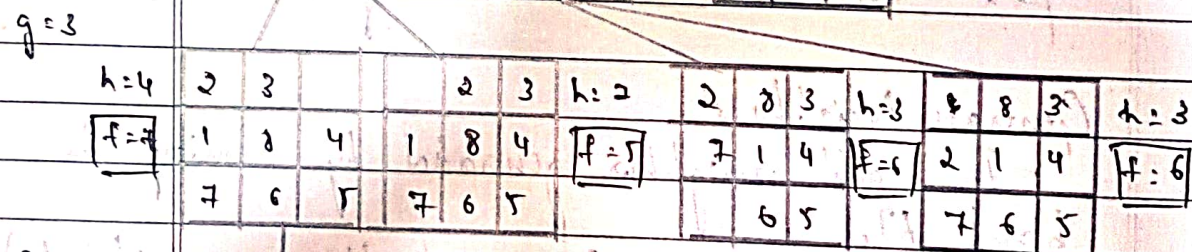
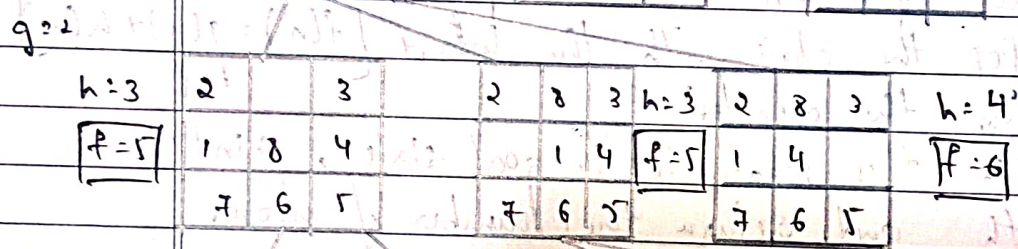
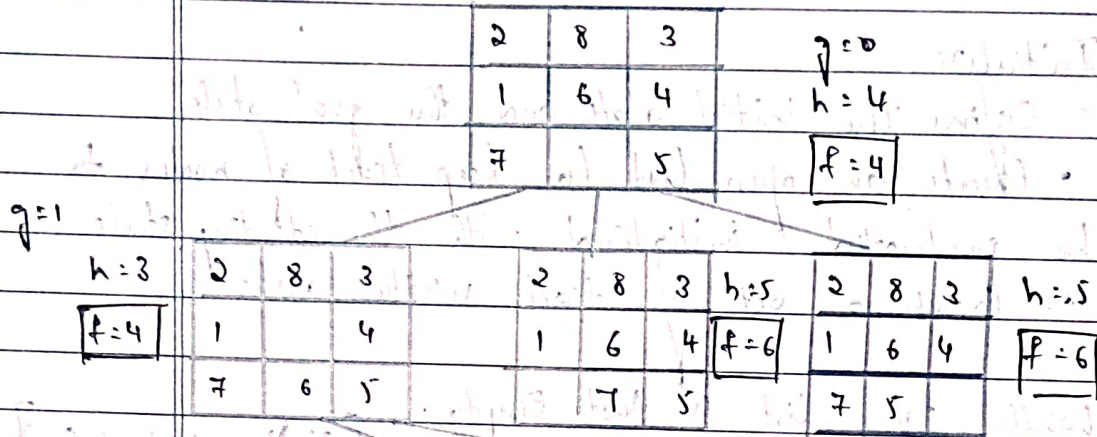
State Space Tree:

2	8	3
1	6	4
7		5

↑

Initial state

Goal state



Final state

II Using Manhattan Distance

1. Initialize

- Define the initial state and the goal state
- Create an open list to keep track of nodes to be evaluated, initialized with the starting state
- Create a set of track visited states.

2. While Open list is Not Empty.

- Pop the state with the lowest $[f(n) = g(n) + h(n)]$ from the open list.
- If the state is the goal state, print the path and return the number of moves.

Generate Neighbours.

- For each neighbor generated by moving the blank tile.
- Calculate $g(n)$ (depth of the current state) $+1$
- Calculate $h(n)$ (sum of manhattan distance from all the tiles).
- Compute $f(n) = g(n) + h(n)$.
- If the neighbor has not been visited, add it to open list and mark it as visited.

3. Sort the openlist based on $f(n)$ for the next iteration.

4. If the open list is empty and no solution is found, return -1.

State Space Tree:-

Initial State:

Goal State:

2	8	3
1	6	4
7		5

1	2	3
8		4
7	6	5

Initial
State

→

2	8	3
1	6	4
7		5

 $g=0$ $h = 1+2+1+1 = 5$ $f=5$ $g=1$ $h=4$

2	8	3
1		4
7	6	5

 $f=5$

2	8	3
1	6	4
7	5	

 $f=7$

2	8	3
	6	4
7	5	

 $f=7$ $g=2$ $h=5$

2	8	3
1	4	
7	6	5

 $f=7$

2	8	3
	1	4
7	6	5

 $f=7$

2		3
1	8	4
7	6	5

 $f=5$ $g=3$ $h=2$

	2	3
1	8	4
7	6	5

 $f=5$ $h=4$

2	3	
1	8	4
7	6	5

 $f=7$ $g=4$ $h=1$

1	2	3
	8	4
7	6	5

 $f=5$ $h=0$

1	2	3
8		4
7	6	5

 $f=5$

↑

Final State

15.10