ITCS 6150 – Intelligent Systems Project Report

Title: 8 Puzzle Problem Solution using A* Search

Name: Sourav Roy Choudhury
ID- 801100959

Date: 02-11-2019

Purpose:
The objective of the project is to solve the 8-puzzle problem using A star (*) search algorithm. In order to determine the best possible ways, two heuristic techniques are discussed:

a. Misplaced Tiles Concept
b. Manhattan Distance Concept

Project Details:

Application Overview:

INPUT:

a. The user is asked to enter numbers from 0 to 8. This is the representation of initial state of puzzle.
b. The user is asked to enter numbers from 0 to 8. This is the representation of final (goal) state of puzzle.
c. The application basically implements various combinations to reach or match the goal state. The process is carried out by moving the tiles present in free space i.e 0.
d. In order to find out the best heuristic the user has a choice of two methods; Manhattan Distance or Misplaced Tiles method.
e. The main objective is to expand the state with least heuristic.
f. The above step is repeated till the goal state is reached.
g. For every move we need to calculate f(n) which depends directly on two separate variables g(n) and h(n). A* algorithm runs on the principle that lowest sum of f(n) decides the next possible best move.
h. g(n)- It is basically defined as the distance from the initial state to the current state.

i. h(n)- It is basically defined as the distance from the current state to goal or final state.
j. Once the puzzle has been solved, the application prints a couple of outputs :

Heuristics:

Manhattan Distance: In this heuristic technique each number in present state is compared to goal state. It is an admissible heuristic distance where the return value is the cumulative sum of steps needs by each tile to reach their individual goal state from current state.

Formula:
[(difference of x coordinates from initial to goal node) + (difference of y coordinates from initial to goal node)]

Misplaced tiles: In this heuristic technique each element of present node is compared with goal node, this is done in order to compute the number of steps taken by the misplaced tile to reach the goal node. The number of misplaced tiles gives us the heuristic value.

Language :

The program was developed in Java

Input:

The user can choose from two heuristics methods i.e Manhattan Distance and Misplaced Tiles. Also the user has to provide the initial state and desired final state as per discretion.

Output:

Path Direction:
This is the direction to goal. The traversal here is represented as Left, Right, up and Down.

Final Count:
This is the calculated total path cost from n to goal i.e from initial state to final state. This is displayed as f(n) which is basically the summation of g(n) and h(n). Kindly note that just the final f(n) is displayed in the output.

Output sequence in matrix format:

EXAMPLE:

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

  ↓

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

Total Nodes explored:
This is total number of nodes explored by the traversal of A* search.

Total Nodes generated:
This is the entire sum of the nodes that are generated i.e the explored nodes and the unexplored nodes together.

Program Structure:

Global Variables:

Nodes_generated - This is a global variable of integer type which holds the value for total nodes generated. It is initialized at 0 and is incremented as and when the nodes are generated and stored in queue.

heuristicChoice - This is a static integer variable. It is basically used to record the user input while deciding the choice of heuristic initially. This is later used as a parameter in the program.

goalStateArray – This is a statically declared variable. The type is an integer array. It's basically used to store the input array sequence from user and compare it to the goal state provided so that if the goal and initial states are one and same, the program comes out and prints the message "Goal State Reached".

Functions:

misplacedTilesusingA (Main Method)–
It is used to take the input parameter for goal and initial state from the system and call the calculatedata method to execute the heuristic function.

calculateData –
This function calls other methods which is used to set the array in a queue to perform the operations, calculate the misplaced tiles heuristic, calculate the state and the number of nodes traversed. Hene it helps in calculating f(n).
The input parameters are goalStateArray and initialStateArray.

findMove –
It is used to call methods that helps in calculating the states up, down, left and right.
The input parameter is a node detail. The return type is an array of node states.

Up,Down Left, Right –
It takes a node detail as input and calculates the new position of blank space (0 has been used to calculate here) by moving it to directions up,down,left,right.
The input parameter is a Plain old java object (pojo) class in java.
The output parameter is the new node object which has been displaced.

misplacedTiles –
It basically calculate the number of misplaced tiles or the number of elelemts in array which are not as per goal state.
The input parameter is a initial goal object.

calculateManhattanDistance –
It calculates the Manhattan distance heuristic for each node and state. I.e the sum of tiles from their goal positions.

The input parameter is an array list.

checkPosition –
It returns the position of the element of from the state_array, which in turn is needed to calculate the absolute value.

Summarization of Results					
Initial State	Manhattan Distance		Misplaced Tiles		Final State
	Nodes Expanded	Nodes Generated	Nodes Expanded	Nodes Generated	
123 745 680	11	21	22	42	123 864 750
281 346 750	7	13	8	15	321 804 756
123 046 758	4	9	4	9	123 456 780
123 804 765	25	44	38	67	281 043 765

TEST CASES

A.i)

```
Enter Heuristic function
****************************
1) Manhattan 2) Misplaced tiles
1
Enter Initial State - Use space after every number (example : 0 1 2 3 4 5 6 7 8):

1 2 3 7 4 5 6 8 0
Enter goal State - Use space after every number (example : 0 1 2 3 4 5 6 7 8) :

1 2 3 8 6 4 7 5 0
[1, 2, 3, 7, 4, 0, 6, 8, 5]
[1, 2, 3, 7, 4, 5, 6, 0, 8]
[1, 2, 0, 7, 4, 3, 6, 8, 5]
[1, 2, 3, 7, 0, 4, 6, 8, 5]
[1, 0, 3, 7, 2, 4, 6, 8, 5]
[1, 2, 3, 7, 8, 4, 6, 0, 5]
[1, 2, 3, 0, 7, 4, 6, 8, 5]
[1, 2, 3, 7, 8, 4, 0, 6, 5]
[1, 2, 3, 7, 8, 4, 6, 5, 0]
[1, 2, 3, 7, 8, 0, 6, 5, 4]
[1, 2, 3, 0, 8, 4, 7, 6, 5]
[1, 2, 0, 7, 8, 3, 6, 5, 4]
[1, 2, 3, 7, 0, 8, 6, 5, 4]
[0, 2, 3, 1, 8, 4, 7, 6, 5]
[1, 2, 3, 8, 0, 4, 7, 6, 5]
[1, 0, 3, 8, 2, 4, 7, 6, 5]
[1, 2, 3, 8, 6, 4, 7, 0, 5]
[1, 2, 3, 8, 4, 0, 7, 6, 5]
[1, 2, 3, 8, 6, 4, 0, 7, 5]
[1, 2, 3, 8, 6, 4, 7, 5, 0]
UP
LEFT
DOWN
LEFT
UP
RIGHT
DOWN
```

```
RIGHT
Final count f(n)8

1       2       3
7       4       5
6       8       0

1       2       3
7       4       0
6       8       5

1       2       3
7       0       4
6       8       5

1       2       3
7       8       4
6       0       5

1       2       3
7       8       4
0       6       5

1       2       3
0       8       4
7       6       5

1       2       3
8       0       4
7       6       5

1       2       3
8       6       4
7       0       5

1       2       3
8       6       4
7       5       0
Total nodes explored :11
Total nodes generated: 21
```

A.ii)

```
Enter Heuristic function
****************************
1) Manhattan 2) Misplaced tiles
2
Enter Initial State - Use space after every number (example : 0 1 2 3 4 5 6 7 8):

1 2 3 7 4 5 6 8 0
Enter goal State - Use space after every number (example : 0 1 2 3 4 5 6 7 8) :

1 2 3 8 6 4 7 5 0
[1, 2, 3, 7, 4, 0, 6, 8, 5]
[1, 2, 3, 7, 4, 5, 6, 0, 8]
[1, 2, 0, 7, 4, 3, 6, 8, 5]
[1, 2, 3, 7, 0, 4, 6, 8, 5]
[1, 2, 3, 7, 0, 5, 6, 4, 8]
[1, 2, 3, 7, 4, 5, 0, 6, 8]
[1, 0, 3, 7, 2, 4, 6, 8, 5]
[1, 2, 3, 7, 8, 4, 6, 0, 5]
[1, 2, 3, 0, 7, 4, 6, 8, 5]
[1, 2, 3, 0, 4, 5, 7, 6, 8]
[0, 2, 3, 1, 7, 4, 6, 8, 5]
[1, 2, 3, 6, 7, 4, 0, 8, 5]
[0, 2, 3, 1, 4, 5, 7, 6, 8]
[1, 2, 3, 4, 0, 5, 7, 6, 8]
[1, 2, 3, 7, 8, 4, 0, 6, 5]
[1, 2, 3, 7, 8, 4, 6, 5, 0]
[1, 0, 3, 7, 2, 5, 6, 4, 8]
[1, 2, 3, 0, 7, 5, 6, 4, 8]
[1, 2, 3, 7, 5, 0, 6, 4, 8]
[1, 2, 3, 7, 8, 0, 6, 5, 4]
[1, 2, 0, 7, 5, 3, 6, 4, 8]
[1, 2, 3, 7, 5, 8, 6, 4, 0]
[0, 1, 3, 7, 2, 4, 6, 8, 5]
[1, 3, 0, 7, 2, 4, 6, 8, 5]
[1, 2, 3, 6, 7, 4, 8, 0, 5]
[1, 2, 3, 0, 8, 4, 7, 6, 5]
```

```
[1, 0, 2, 7, 4, 3, 6, 8, 5]
[0, 2, 3, 1, 7, 5, 6, 4, 8]
[1, 2, 3, 6, 7, 5, 0, 4, 8]
[1, 0, 3, 4, 2, 5, 7, 6, 8]
[1, 2, 3, 4, 6, 5, 7, 0, 8]
[1, 2, 3, 4, 5, 0, 7, 6, 8]
[0, 2, 3, 1, 8, 4, 7, 6, 5]
[1, 2, 3, 8, 0, 4, 7, 6, 5]
[1, 2, 3, 4, 6, 5, 0, 7, 8]
[1, 2, 3, 4, 6, 5, 7, 8, 0]
[1, 0, 3, 8, 2, 4, 7, 6, 5]
[1, 2, 3, 8, 6, 4, 7, 0, 5]
[1, 2, 3, 8, 4, 0, 7, 6, 5]
[1, 2, 3, 8, 6, 4, 0, 7, 5]
[1, 2, 3, 8, 6, 4, 7, 5, 0]
UP
LEFT
DOWN
LEFT
UP
RIGHT
DOWN
RIGHT
Final count f(n)8

1       2       3
7       4       5
6       8       0

1       2       3
7       4       0
6       8       5

1       2       3
7       0       4
6       8       5

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

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

1       2       3
0       8       4
7       6       5

1       2       3
8       0       4
7       6       5

1       2       3
8       6       4
7       0       5

1       2       3
8       6       4
7       5       0
Total nodes explored :22
Total nodes generated: 42
```

B. i)

```
2. Enter Heuristic function
****************************
1) Manhattan 2) Misplaced tiles
1
Enter Initial State - Use space after every number (example : 0 1 2 3 4 5 6 7 8):

2 8 1 3 4 6 7 5 0
Enter goal State - Use space after every number (example : 0 1 2 3 4 5 6 7 8) :

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

```
[2, 8, 1, 3, 4, 0, 7, 5, 6]
[2, 8, 1, 3, 4, 6, 7, 0, 5]
[2, 8, 0, 3, 4, 1, 7, 5, 6]
[2, 8, 1, 3, 0, 4, 7, 5, 6]
[2, 0, 1, 3, 8, 4, 7, 5, 6]
[2, 8, 1, 3, 5, 4, 7, 0, 6]
[2, 8, 1, 0, 3, 4, 7, 5, 6]
[0, 2, 1, 3, 8, 4, 7, 5, 6]
[2, 1, 0, 3, 8, 4, 7, 5, 6]
[3, 2, 1, 0, 8, 4, 7, 5, 6]
[3, 2, 1, 7, 8, 4, 0, 5, 6]
[3, 2, 1, 8, 0, 4, 7, 5, 6]
UP
LEFT
UP
LEFT
DOWN
RIGHT
Final count f(n)6

2	8	1
3	4	6
7	5	0

2	8	1
3	4	0
7	5	6

2	8	1
3	0	4
7	5	6

2	0	1
3	8	4
7	5	6

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

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

3       2       1
8       0       4
7       5       6
Total nodes explored :7
Total nodes generated: 13
```

B. ii)

```
Enter Heuristic function
****************************
1) Manhattan 2) Misplaced tiles
2
Enter Initial State - Use space after every number (example : 0 1 2 3 4 5 6 7 8):

2 8 1 3 4 6 7 5 0
Enter goal State - Use space after every number (example : 0 1 2 3 4 5 6 7 8) :

3 2 1 8 0 4 7 5 6
[2, 8, 1, 3, 4, 0, 7, 5, 6]
[2, 8, 1, 3, 4, 6, 7, 0, 5]
[2, 8, 0, 3, 4, 1, 7, 5, 6]
[2, 8, 1, 3, 0, 4, 7, 5, 6]
[2, 0, 1, 3, 8, 4, 7, 5, 6]
[2, 8, 1, 3, 5, 4, 7, 0, 6]
[2, 8, 1, 0, 3, 4, 7, 5, 6]
[0, 2, 1, 3, 8, 4, 7, 5, 6]
[2, 1, 0, 3, 8, 4, 7, 5, 6]
[0, 8, 1, 2, 3, 4, 7, 5, 6]
[2, 8, 1, 7, 3, 4, 0, 5, 6]
[3, 2, 1, 0, 8, 4, 7, 5, 6]
[3, 2, 1, 7, 8, 4, 0, 5, 6]
[3, 2, 1, 8, 0, 4, 7, 5, 6]
UP
LEFT
```

```
UP
LEFT
DOWN
RIGHT
Final count f(n)6

2       8       1
3       4       6
7       5       0

2       8       1
3       4       0
7       5       6

2       8       1
3       0       4
7       5       6

2       0       1
3       8       4
7       5       6

0       2       1
3       8       4
7       5       6

3       2       1
0       8       4
7       5       6

3       2       1
8       0       4
7       5       6
Total nodes explored :8
Total nodes generated: 15
```

C.i)

```
Enter Heuristic function
****************************
1) Manhattan 2) Misplaced tiles
1
Enter Initial State - Use space after every number (example : 0 1 2 3 4 5 6 7 8):

1 2 3 0 4 6 7 5 8
Enter goal State - Use space after every number (example : 0 1 2 3 4 5 6 7 8) :

1 2 3 4 5 6 7 8 0
[0, 2, 3, 1, 4, 6, 7, 5, 8]
[1, 2, 3, 7, 4, 6, 0, 5, 8]
[1, 2, 3, 4, 0, 6, 7, 5, 8]
[1, 0, 3, 4, 2, 6, 7, 5, 8]
[1, 2, 3, 4, 5, 6, 7, 0, 8]
[1, 2, 3, 4, 6, 0, 7, 5, 8]
[1, 2, 3, 4, 5, 6, 0, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 8, 0]
RIGHT
DOWN
RIGHT
Final count f(n)3

1	2	3
0	4	6
7	5	8

1	2	3
4	0	6
7	5	8

1	2	3
4	5	6
7	0	8

1	2	3
4	5	6
7	8	0
Total nodes explored :4
Total nodes generated: 9
```

C.ii)

```
Enter Heuristic function
****************************
1) Manhattan 2) Misplaced tiles
2
Enter Initial State - Use space after every number (example : 0 1 2 3 4 5 6 7 8):

1 2 3 0 4 6 7 5 8
Enter goal State - Use space after every number (example : 0 1 2 3 4 5 6 7 8) :

1 2 3 4 5 6 7 8 0
[0, 2, 3, 1, 4, 6, 7, 5, 8]
[1, 2, 3, 7, 4, 6, 0, 5, 8]
[1, 2, 3, 4, 0, 6, 7, 5, 8]
[1, 0, 3, 4, 2, 6, 7, 5, 8]
[1, 2, 3, 4, 5, 6, 7, 0, 8]
[1, 2, 3, 4, 6, 0, 7, 5, 8]
[1, 2, 3, 4, 5, 6, 0, 7, 8]
[1, 2, 3, 4, 5, 6, 7, 8, 0]
RIGHT
DOWN
RIGHT
Final count f(n)3

1       2       3
0       4       6
7       5       8

1       2       3
4       0       6
7       5       8

1       2       3
4       5       6
7       0       8

1       2       3
```

```
4         5         6
7         8         0
Total nodes explored :4
Total nodes generated: 9
```

D. i)

```
Enter Heuristic function
****************************
1) Manhattan 2) Misplaced tiles
1
Enter Initial State - Use space after every number (example : 0 1 2 3 4 5 6 7 8):

1 2 3 8 0 4 7 6 5
Enter goal State - Use space after every number (example : 0 1 2 3 4 5 6 7 8) :

2 8 1 0 4 3 7 6 5
[1, 0, 3, 8, 2, 4, 7, 6, 5]
[1, 2, 3, 8, 6, 4, 7, 0, 5]
[1, 2, 3, 0, 8, 4, 7, 6, 5]
[1, 2, 3, 8, 4, 0, 7, 6, 5]
[0, 2, 3, 1, 8, 4, 7, 6, 5]
[1, 2, 3, 7, 8, 4, 0, 6, 5]
[1, 2, 0, 8, 4, 3, 7, 6, 5]
[1, 2, 3, 8, 4, 5, 7, 6, 0]
[1, 2, 3, 7, 8, 4, 6, 0, 5]
[2, 0, 3, 1, 8, 4, 7, 6, 5]
[1, 0, 2, 8, 4, 3, 7, 6, 5]
[2, 8, 3, 1, 0, 4, 7, 6, 5]
[2, 3, 0, 1, 8, 4, 7, 6, 5]
[2, 8, 3, 1, 6, 4, 7, 0, 5]
[2, 8, 3, 0, 1, 4, 7, 6, 5]
[2, 8, 3, 1, 4, 0, 7, 6, 5]
[0, 8, 3, 2, 1, 4, 7, 6, 5]
[2, 8, 3, 7, 1, 4, 0, 6, 5]
[2, 8, 0, 1, 4, 3, 7, 6, 5]
[2, 8, 3, 1, 4, 5, 7, 6, 0]
[8, 0, 3, 2, 1, 4, 7, 6, 5]
[2, 8, 3, 7, 1, 4, 6, 0, 5]
```

```
[2, 0, 8, 1, 4, 3, 7, 6, 5]
[2, 4, 8, 1, 0, 3, 7, 6, 5]
[0, 2, 8, 1, 4, 3, 7, 6, 5]
[2, 4, 8, 1, 6, 3, 7, 0, 5]
[2, 4, 8, 0, 1, 3, 7, 6, 5]
[2, 4, 8, 1, 3, 0, 7, 6, 5]
[0, 4, 8, 2, 1, 3, 7, 6, 5]
[2, 4, 8, 7, 1, 3, 0, 6, 5]
[1, 2, 8, 0, 4, 3, 7, 6, 5]
[1, 2, 8, 7, 4, 3, 0, 6, 5]
[1, 2, 8, 4, 0, 3, 7, 6, 5]
[4, 0, 8, 2, 1, 3, 7, 6, 5]
[1, 0, 8, 4, 2, 3, 7, 6, 5]
[1, 2, 8, 4, 6, 3, 7, 0, 5]
[1, 2, 8, 4, 3, 0, 7, 6, 5]
[2, 4, 8, 7, 1, 3, 6, 0, 5]
[1, 2, 8, 7, 4, 3, 6, 0, 5]
[2, 4, 8, 7, 0, 3, 6, 1, 5]
[2, 4, 8, 7, 1, 3, 6, 5, 0]
[1, 2, 8, 7, 0, 3, 6, 4, 5]
[1, 2, 8, 7, 4, 3, 6, 5, 0]
[2, 8, 3, 1, 4, 5, 7, 0, 6]
[1, 4, 2, 8, 0, 3, 7, 6, 5]
[0, 1, 2, 8, 4, 3, 7, 6, 5]
[8, 1, 2, 0, 4, 3, 7, 6, 5]
[8, 1, 2, 7, 4, 3, 0, 6, 5]
[8, 1, 2, 4, 0, 3, 7, 6, 5]
[8, 1, 2, 7, 4, 3, 6, 0, 5]
[8, 0, 2, 4, 1, 3, 7, 6, 5]
[8, 1, 2, 4, 6, 3, 7, 0, 5]
[8, 1, 2, 4, 3, 0, 7, 6, 5]
[8, 1, 2, 7, 0, 3, 6, 4, 5]
[8, 1, 2, 7, 4, 3, 6, 5, 0]
[8, 1, 3, 2, 0, 4, 7, 6, 5]
[8, 3, 0, 2, 1, 4, 7, 6, 5]
[8, 1, 3, 2, 6, 4, 7, 0, 5]
[8, 1, 3, 0, 2, 4, 7, 6, 5]
[8, 1, 3, 2, 4, 0, 7, 6, 5]
[0, 1, 3, 8, 2, 4, 7, 6, 5]
[8, 1, 3, 7, 2, 4, 0, 6, 5]
```

```
[8, 1, 0, 2, 4, 3, 7, 6, 5]
[8, 1, 3, 2, 4, 5, 7, 6, 0]
[8, 0, 1, 2, 4, 3, 7, 6, 5]
[8, 4, 1, 2, 0, 3, 7, 6, 5]
[0, 8, 1, 2, 4, 3, 7, 6, 5]
[2, 8, 1, 0, 4, 3, 7, 6, 5]
LEFT
UP
RIGHT
DOWN
LEFT
UP
RIGHT
DOWN
RIGHT
UP
LEFT
LEFT
DOWN
Final count f(n)13

1       2       3
8       0       4
7       6       5

1       2       3
0       8       4
7       6       5

0       2       3
1       8       4
7       6       5

2       0       3
1       8       4
7       6       5

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

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

0       8       3
2       1       4
7       6       5

8       0       3
2       1       4
7       6       5

8       1       3
2       0       4
7       6       5

8       1       3
2       4       0
7       6       5

8       1       0
2       4       3
7       6       5

8       0       1
2       4       3
7       6       5

0       8       1
2       4       3
7       6       5

2       8       1
0       4       3
7       6       5
Total nodes explored :39
Total nodes generated: 69
```

D. ii)

```
Enter Heuristic function
*****************************
1) Manhattan 2) Misplaced tiles
2
Enter Initial State - Use space after every number (example : 0 1 2 3 4 5 6 7 8):

1 2 3 8 0 4 7 6 5
Enter goal State - Use space after every number (example : 0 1 2 3 4 5 6 7 8) :

2 8 1 0 4 3 7 6 5
[1, 0, 3, 8, 2, 4, 7, 6, 5]
[1, 2, 3, 8, 6, 4, 7, 0, 5]
[1, 2, 3, 0, 8, 4, 7, 6, 5]
[1, 2, 3, 8, 4, 0, 7, 6, 5]
[1, 2, 0, 8, 4, 3, 7, 6, 5]
[1, 2, 3, 8, 4, 5, 7, 6, 0]
[1, 0, 2, 8, 4, 3, 7, 6, 5]
[0, 1, 3, 8, 2, 4, 7, 6, 5]
[1, 3, 0, 8, 2, 4, 7, 6, 5]
[1, 4, 2, 8, 0, 3, 7, 6, 5]
[0, 1, 2, 8, 4, 3, 7, 6, 5]
[0, 2, 3, 1, 8, 4, 7, 6, 5]
[1, 2, 3, 7, 8, 4, 0, 6, 5]
[8, 1, 2, 0, 4, 3, 7, 6, 5]
[1, 2, 3, 8, 4, 5, 7, 0, 6]
[1, 2, 3, 8, 6, 4, 0, 7, 5]
[1, 2, 3, 8, 6, 4, 7, 5, 0]
[8, 1, 3, 0, 2, 4, 7, 6, 5]
[1, 3, 4, 8, 2, 0, 7, 6, 5]
[2, 0, 3, 1, 8, 4, 7, 6, 5]
[2, 8, 3, 1, 0, 4, 7, 6, 5]
[2, 3, 0, 1, 8, 4, 7, 6, 5]
[2, 8, 3, 1, 6, 4, 7, 0, 5]
[2, 8, 3, 0, 1, 4, 7, 6, 5]
[2, 8, 3, 1, 4, 0, 7, 6, 5]
[2, 8, 0, 1, 4, 3, 7, 6, 5]
[2, 8, 3, 1, 4, 5, 7, 6, 0]
[2, 0, 8, 1, 4, 3, 7, 6, 5]
[1, 3, 4, 8, 2, 5, 7, 6, 0]
```

```
[1, 3, 4, 8, 0, 2, 7, 6, 5]
[1, 2, 3, 7, 8, 4, 6, 0, 5]
[8, 1, 2, 7, 4, 3, 0, 6, 5]
[8, 1, 2, 4, 0, 3, 7, 6, 5]
[0, 8, 3, 2, 1, 4, 7, 6, 5]
[2, 8, 3, 7, 1, 4, 0, 6, 5]
[8, 1, 3, 7, 2, 4, 0, 6, 5]
[8, 1, 3, 2, 0, 4, 7, 6, 5]
[2, 3, 4, 1, 8, 0, 7, 6, 5]
[1, 4, 2, 8, 6, 3, 7, 0, 5]
[1, 4, 2, 0, 8, 3, 7, 6, 5]
[1, 4, 2, 8, 3, 0, 7, 6, 5]
[2, 3, 4, 1, 8, 5, 7, 6, 0]
[2, 3, 4, 1, 0, 8, 7, 6, 5]
[2, 8, 3, 1, 4, 5, 7, 0, 6]
[1, 2, 3, 0, 6, 4, 8, 7, 5]
[8, 0, 3, 2, 1, 4, 7, 6, 5]
[8, 1, 3, 2, 6, 4, 7, 0, 5]
[8, 1, 3, 2, 4, 0, 7, 6, 5]
[1, 2, 3, 8, 0, 5, 7, 4, 6]
[1, 2, 3, 8, 4, 5, 0, 7, 6]
[8, 1, 0, 2, 4, 3, 7, 6, 5]
[8, 1, 3, 2, 4, 5, 7, 6, 0]
[0, 4, 2, 1, 8, 3, 7, 6, 5]
[1, 4, 2, 7, 8, 3, 0, 6, 5]
[1, 0, 4, 8, 3, 2, 7, 6, 5]
[1, 3, 4, 8, 6, 2, 7, 0, 5]
[1, 3, 4, 0, 8, 2, 7, 6, 5]
[2, 8, 3, 1, 6, 4, 0, 7, 5]
[2, 8, 3, 1, 6, 4, 7, 5, 0]
[2, 4, 8, 1, 0, 3, 7, 6, 5]
[0, 2, 8, 1, 4, 3, 7, 6, 5]
[8, 0, 1, 2, 4, 3, 7, 6, 5]
[1, 2, 3, 8, 6, 0, 7, 5, 4]
[8, 4, 1, 2, 0, 3, 7, 6, 5]
[0, 8, 1, 2, 4, 3, 7, 6, 5]
[2, 8, 1, 0, 4, 3, 7, 6, 5]
UP
LEFT
DOWN
```

```
RIGHT
RIGHT
UP
LEFT
LEFT
DOWN
Final count f(n)9

1       2       3
8       0       4
7       6       5

1       0       3
8       2       4
7       6       5

0       1       3
8       2       4
7       6       5

8       1       3
0       2       4
7       6       5

8       1       3
2       0       4
7       6       5

8       1       3
2       4       0
7       6       5

8       1       0
2       4       3
7       6       5

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

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

2       8       1
0       4       3
7       6       5
Total nodes explored :38
Total nodes generated: 67
```