CS571 - ARTIFICIAL INTELLIGENCE LAB

ASSIGNMENT-2: A* Search

Intaj Ahmed Choudhury – 2211MC09
Ankit Anand – 2311MC04
Khushbu Bharti – 2311MC21

QUESTION :

1. In a general search algorithm each state (n) maintains a function f(n) = g(n) + h(n) where g(n) is the least cost from source state to state n found so far and h(n) is the estimated cost of the optimal path from state n to the goal state. Implement a search algorithm for solving the 8-puzzle problem with following assumptions.

A. g(n) = least cost from source state to current state so far.

B. Heuristics

   a. h1(n) = 0.
   b. h2(n) = number of tiles displaced from their destined position.
   c. h3(n) = sum of Manhattan distance of each tile from the goal position.
   d. h4(n) = Devise a heuristic such that $h(n) > h*(n)$.

Solution

Algorithm :

STEP 1 : Take the initial state of the puzzle from user. Target state is fixed.

STEP 2 : Check whether the puzzle is solvable or not by counting number of inversion.

STEP 3 : If puzzle is Solvable. We initialize the constructer of class "state" with parameters value, level and fx and create object for initial state.

STEP 4 : Taking input from user for the heuristic (hx) function. According to the input we are calculating the fx value. We are taking two lists open and close. The states which are visited and yet to be explored are kept in open list and the explored states are kept in closed list.

STEP 5 : Check the fx value for all the states in the open list. State with the minimum fx value is then explored and after exploration it is put into closed list. Step 4 and step 5 are repeated until the target state is found

Observation:

1. We have got that better heuristics is Manhattan Distance of each tiles from the goal.
2. We have observed that the states which are expanded by using better heuristics (Manhattan Distance) are also in the other solutions where we use a. heuristics function value as zero, b. heuristics function as misplaced tiles, c. heuristics function as Euclidean distance.
3. We have observed and verified the monotonic restriction for each heuristics.
4. We have observed un-reachability for the input of 8-puzzle by using problem is solvable otherwise non-solvable or un-reachable.

Case 1: hx = 0

```
Enter Initial State :
1 2 3
-1 4 6
7 5 8

-------Initial State-------

[1, 2, 3]
[-1, 4, 6]
[7, 5, 8]


-------Target State-------

[1, 2, 3]
[4, 5, 6]
[7, 8, -1]



--------Choose heuristic : -------

1. Depth of node

2. Missplaced Tiles

3. Manhattan Distance

4. Nilsson's sequence score

1



Length of Optimal path :  3

[1, 2, 3]
[-1, 4, 6]
[7, 5, 8]

[1, 2, 3]
[4, -1, 6]
[7, 5, 8]

[1, 2, 3]
[4, 5, 6]
[7, -1, 8]

[1, 2, 3]
[4, 5, 6]
[7, 8, -1]
No. of steps explored = 13
Total execution time in minute :
  0.03329644694444445
```

Case 2: hx = No. of misplaced tiles

```
Enter Initial State :
1 2 3
-1 4 6
7 5 8

-------Initial State-------

[1, 2, 3]
[-1, 4, 6]
[7, 5, 8]


-------Target State-------

[1, 2, 3]
[4, 5, 6]
[7, 8, -1]



--------Choose heuristic : -------

1. Depth of node

2. Missplaced Tiles

3. Manhattan Distance

4. Nilsson's sequence score

2



Length of Optimal path :  3

[1, 2, 3]
[-1, 4, 6]
[7, 5, 8]

[1, 2, 3]
[4, -1, 6]
[7, 5, 8]

[1, 2, 3]
[4, 5, 6]
[7, -1, 8]

[1, 2, 3]
[4, 5, 6]
[7, 8, -1]
No. of steps explored = 3
Total execution time in minute :
  0.10459019777777778
```

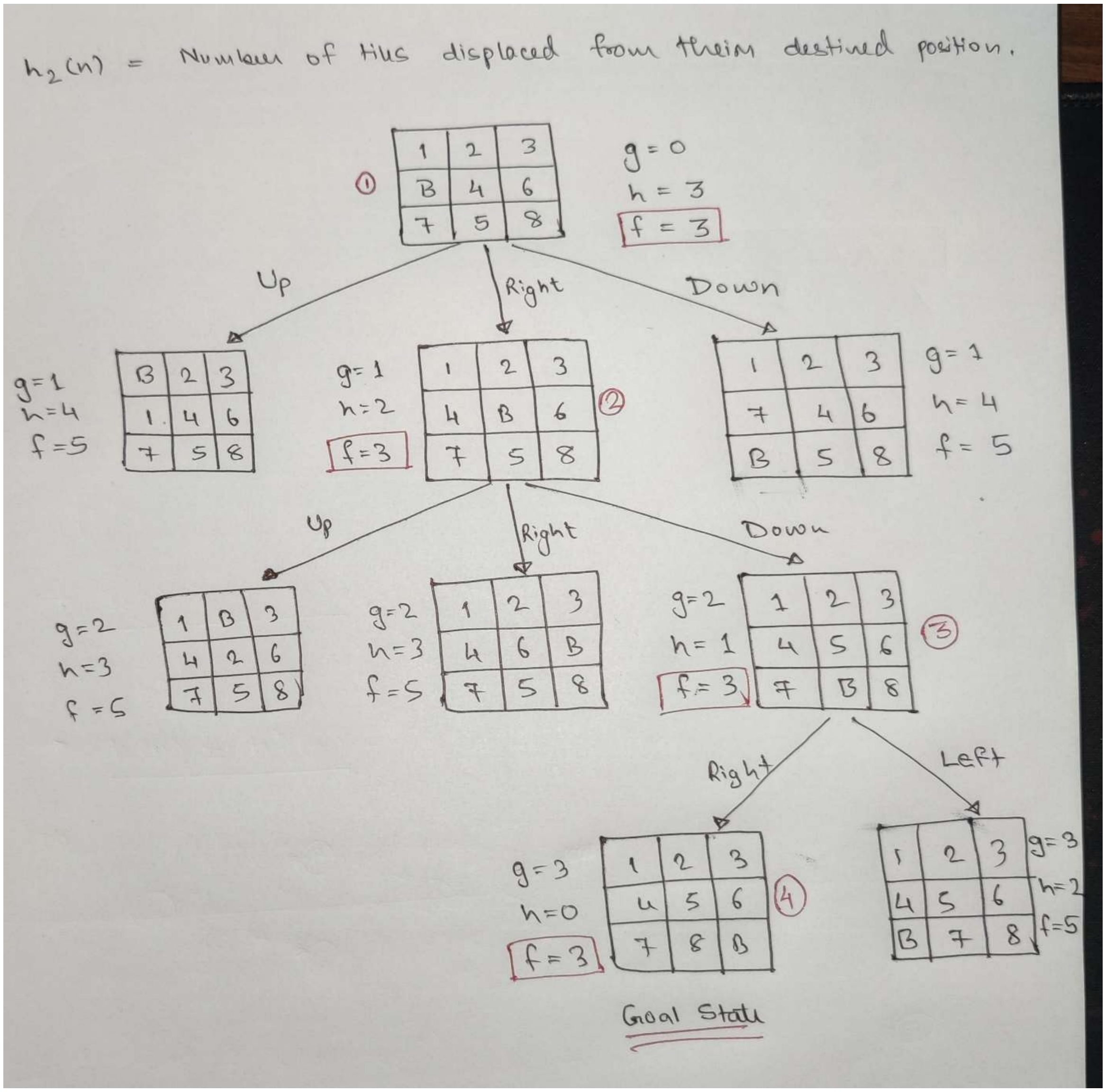
Case 3: hx = Manhattan distsnce

```
Enter Initial State :
1 2 3
-1 4 6
7 5 8

-------Initial State-------

[1, 2, 3]
[-1, 4, 6]
[7, 5, 8]


-------Target State-------

[1, 2, 3]
[4, 5, 6]
[7, 8, -1]



--------Choose heuristic : -------

1. Depth of node

2. Missplaced Tiles

3. Manhattan Distance

4. Nilsson's sequence score

3



Length of Optimal path :  3

[1, 2, 3]
[-1, 4, 6]
[7, 5, 8]

[1, 2, 3]
[4, -1, 6]
[7, 5, 8]

[1, 2, 3]
[4, 5, 6]
[7, -1, 8]

[1, 2, 3]
[4, 5, 6]
[7, 8, -1]
No. of steps explored = 3
Total execution time in minute :
  0.03913056833333333
```

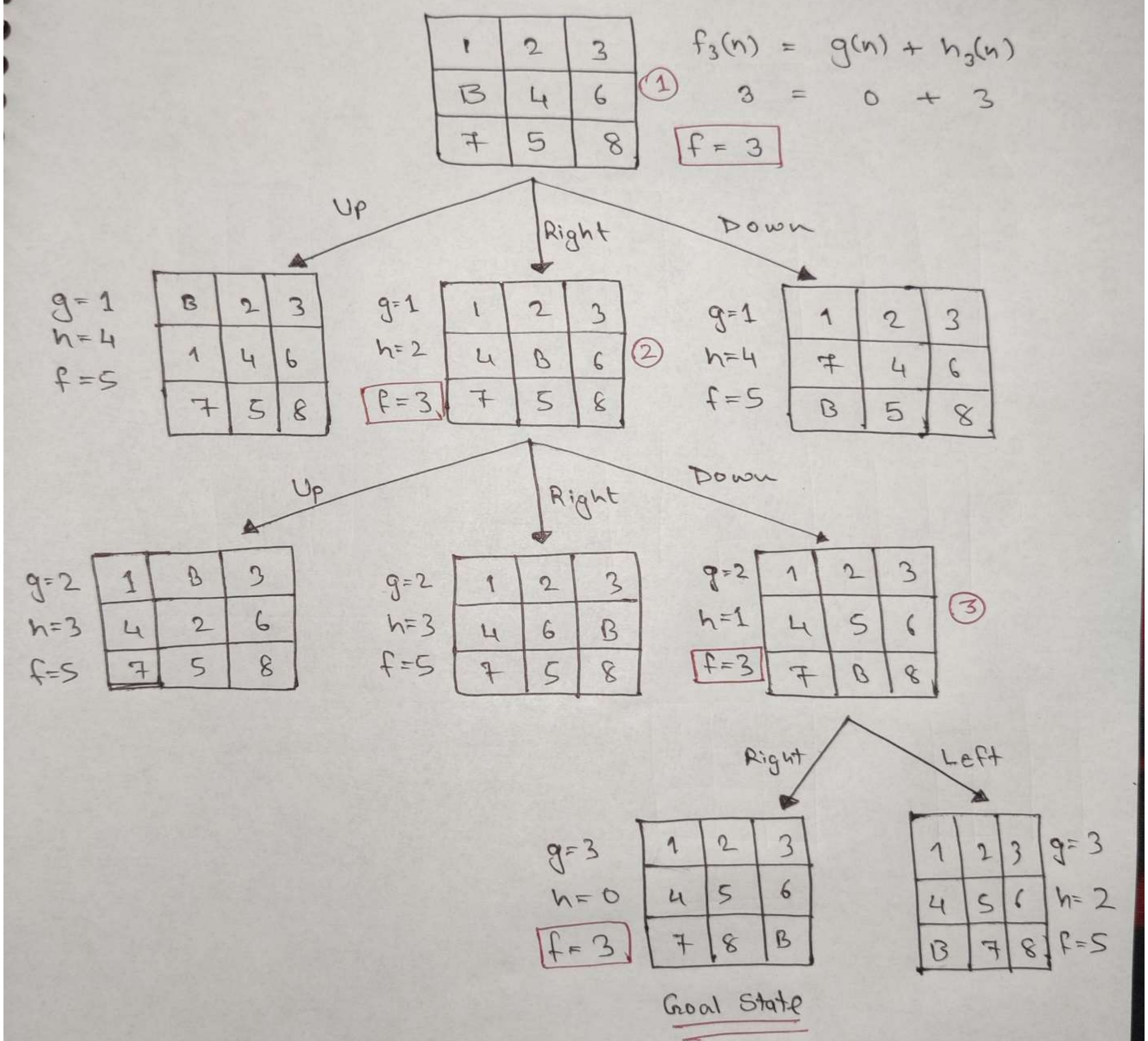
Case 4: hx = Nilsson's sequence score

```
Enter Initial State :
1 2 3
-1 4 6
7 5 8

-------Initial State-------

[1, 2, 3]
[-1, 4, 6]
[7, 5, 8]


-------Target State-------

[1, 2, 3]
[4, 5, 6]
[7, 8, -1]



--------Choose heuristic : -------

1. Depth of node

2. Missplaced Tiles

3. Manhattan Distance

4. Nilsson's sequence score

4



Length of Optimal path :  3

[1, 2, 3]
[-1, 4, 6]
[7, 5, 8]

[1, 2, 3]
[4, -1, 6]
[7, 5, 8]

[1, 2, 3]
[4, 5, 6]
[7, -1, 8]

[1, 2, 3]
[4, 5, 6]
[7, 8, -1]
No. of steps explored = 21
Total execution time in minute :
  0.020253619444444444
```

2.. Observe and verify that all the states expanded by better heuristics should also be expanded by inferior heuristics.



$h_3(n)$ = Sum of Manhattan distance of each tile from the goal position



4. Observe un-reachability and provide a proof.

```
Enter Initial State :
1 2 3
4 6 5
7 8 -1

-------Initial State-------

[1, 2, 3]
[4, 6, 5]
[7, 8, -1]


-------Target State-------

[1, 2, 3]
[4, 5, 6]
[7, 8, -1]



problem cannot be solved
Number of steps before termination = 362880
```

For insolvability , we are calculating inversion count of each number in the initial state and adding it . If the total sum is even the problem is solvable but if the sum is odd the problem is unsolvable.

```python
def isSolvable(a):
    arr=[]
    for i in range(3):
        for j in range(3):
            arr.append(a[i][j])
    inv_count = 0
    empty_value = -1
    for i in range(0, 9):
        for j in range(i + 1, 9):
            if arr[j] != empty_value and arr[i] != empty_value and arr[i] > arr[j]:
                inv_count += 1
    if(inv_count % 2):
        return False
    else:
        return True
```

3. Observe and verify monotone restriction on the heuristics.

$h(n) \leq h'(n) + cost(n, n')$

Checking this condition for all the states in optimal path

(h_3 is taken as heuristic)

$n \rightarrow$ parent

$n' \rightarrow$ child

1	2	3
4	5	6
7	8	B

1	2	3
4	6	5
8	7	B

↓

1	2	3
4	6	5
8	B	7

$4 \leq 5+1$

↓

1	2	3
4	B	5
8	6	7

$5 \leq 6+1$

↓

1	2	3
4	5	B
8	6	7

$6 \leq 5+1$

↓

1	2	3
4	5	7
8	6	B

$5 \leq 6+1$

↓

1	2	3
4	5	7
8	B	6

$6 \leq 5+1$

↓

1	2	3
4	5	7
B	8	6

$5 \leq 4+1$

↓

1	2	3
B	5	7
4	8	6

$4 \leq 5+1$

↓

1	2	3
5	B	7
4	8	6

$5 \leq 6+1$

↓

1	2	3
5	7	B
4	8	6

$6 \leq 5+1$

↓

1	2	3
5	7	6
4	8	B

$5 \leq 4+1$

↓

1	2	3
5	7	6
4	B	8

$4 \leq 5+1$

↓

1	2	3
5	B	6
4	7	8

$5 \leq 4+1$

↓

1	2	3
B	5	6
4	7	8

$4 \leq 3+1$

↓

1	2	3
4	5	6
B	7	8

$3 \leq 2+1$

↓

1	2	3
4	5	6
7	B	8

$2 \leq 1+1$

↓

1	2	3
4	5	6
7	8	B

$1 \leq 0+1$