Artificial Intelligence Programming Assignment 1

Name: Sree Vandana

1. **Problem Solving as Search**

In this assignment both Best-first and A* algorithms are implemented to solve the 8-puzzle. For BFS the search stratagy is defined by picking and expanding the lowest cost node first. A* is widly known as the form of best-first search, where evaluation function is combination of both g(n) and h(n).

Evaluation functions:
   i. BFS: f(n) = h(n)
   ii. A* : f(n) = h(n) + g(n)

Heuristic functions:

   i. **Misplaced tile:** this indicates the total number of misplaced tiles in given puzzel.
   ii. **Manhattan Distance:** it is the sum of the distances (vertical and horizontal movements) of the tiles from current location to their goal state.
   iii. **Maximum Metric Distance:** this measures the distance between two points as the maximum difference over any of their axis values. In a 2D grid, for instance, if we have two points (x1, y1), and (x2, y2), the Chebyshev distance between is *max(y2 - y1, x2 - x1)* . this is also known as Chebyshev distance.

5 puzzles initial states used: (blank 'b' is replaced with integer 0)

Puzzles are represented using 3x3 matrix.

   i.   [[4, 5, 0], [6, 1, 8], [7, 3, 2]]
   ii.   [[1, 2, 5], [0, 4, 3], [6, 8, 7]]
   iii.   [[0, 1, 2], [3, 4, 5], [6, 7, 8]]
   iv.   [[1, 2, 3], [4, 0, 5], [6, 7, 8]]
   v.   [[1, 3, 0], [4, 2, 5], [7, 8, 6]]

Goal State:

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

1. **Best first search**

Heuristic 1: Number of misplaced tiles

   i.   **Initial-state: [[4, 5, 0], [6, 1, 8], [7, 3, 2]] → (4 5 0 6 1 8 7 3 2)**
         Could not find solution

ii. **Initial-state: [[1, 2, 5], [0, 4, 3], [6, 8, 7]] → (1 2 5 0 4 3 6 8 7)**
Puzzle solved.
Number of steps: 45

(1 2 5 4 0 3 6 8 7) --> (1 0 5 4 2 3 6 8 7) --> (1 5 0 4 2 3 6 8 7) --> (1 5 3 4 2 0 6 8 7) -
-> (1 5 3 4 2 7 6 8 0) --> (1 5 3 4 2 7 6 0 8) --> (1 5 3 4 0 7 6 2 8) --> (1 0 3 4 5 7 6 2 8
) --> (1 3 0 4 5 7 6 2 8) --> (1 3 7 4 5 0 6 2 8) --> (1 3 7 4 0 5 6 2 8) --> (1 3 7 4 2 5 6
0 8) --> (1 3 7 4 2 5 6 8 0) --> (1 3 7 4 2 0 6 8 5) --> (1 3 0 4 2 7 6 8 5) --> (1 0 3 4 2
7 6 8 5) --> (1 2 3 4 0 7 6 8 5) --> (1 2 3 4 7 0 6 8 5) --> (1 2 3 4 7 5 6 8 0) --> (1 2 3
4 7 5 6 0 8) --> (1 2 3 4 7 5 0 6 8) --> (1 2 3 0 7 5 4 6 8) --> (1 2 3 7 0 5 4 6 8) --> (1
2 3 7 6 5 4 0 8) --> (1 2 3 7 6 5 4 8 0) --> (1 2 3 7 6 0 4 8 5) --> (1 2 3 7 0 6 4 8 5) -->
(1 2 3 0 7 6 4 8 5) --> (1 2 3 4 7 6 0 8 5) --> (1 2 3 4 7 6 8 0 5) --> (1 2 3 4 7 6 8 5 0) -
-> (1 2 3 4 7 0 8 5 6) --> (1 2 3 4 0 7 8 5 6) --> (1 2 3 4 5 7 8 0 6) --> (1 2 3 4 5 7 0 8 6
) --> (1 2 3 0 5 7 4 8 6) --> (1 2 3 5 0 7 4 8 6) --> (1 2 3 5 7 0 4 8 6) --> (1 2 3 5 7 6 4
8 0) --> (1 2 3 5 7 6 4 0 8) --> (1 2 3 5 0 6 4 7 8) --> (1 2 3 0 5 6 4 7 8) --> (1 2 3 4 5
6 0 7 8) --> (1 2 3 4 5 6 7 0 8) --> (1 2 3 4 5 6 7 8 0)

iii. **Initial state: [[0, 1, 2], [3, 4, 5], [6, 7, 8]] → (0 1 2 3 4 5 6 7 8)**
Puzzle Solved
Number of steps: 30
(1 0 2 3 4 5 6 7 8) --> (1 2 0 3 4 5 6 7 8) --> (1 2 5 3 4 0 6 7 8) --> (1 2 5 3 0 4 6 7 8) -
-> (1 2 5 0 3 4 6 7 8) --> (1 2 5 6 3 4 0 7 8) --> (1 2 5 6 3 4 7 0 8) --> (1 2 5 6 3 4 7 8 0
) --> (1 2 5 6 3 0 7 8 4) --> (1 2 5 6 0 3 7 8 4) --> (1 2 5 0 6 3 7 8 4) --> (0 2 5 1 6 3 7
8 4) --> (2 0 5 1 6 3 7 8 4) --> (2 5 0 1 6 3 7 8 4) --> (2 5 3 1 6 0 7 8 4) --> (2 5 3 1 0
6 7 8 4) --> (2 0 3 1 5 6 7 8 4) --> (0 2 3 1 5 6 7 8 4) --> (1 2 3 0 5 6 7 8 4) --> (1 2 3
7 5 6 0 8 4) --> (1 2 3 7 5 6 8 0 4) --> (1 2 3 7 5 6 8 4 0) --> (1 2 3 7 5 0 8 4 6) --> (1
2 3 7 0 5 8 4 6) --> (1 2 3 7 4 5 8 0 6) --> (1 2 3 7 4 5 0 8 6) --> (1 2 3 0 4 5 7 8 6) -->
(1 2 3 4 0 5 7 8 6) --> (1 2 3 4 5 0 7 8 6) --> (1 2 3 4 5 6 7 8 0)

iv. **Initial-state: [[1, 2, 3], [4, 0, 5], [6, 7, 8]] → (1 2 3 4 0 5 6 7 8)**
Puzzle Solved
Number of steps: 26
(1 2 3 4 7 5 6 0 8) --> (1 2 3 4 7 5 0 6 8) --> (1 2 3 0 7 5 4 6 8) --> (1 2 3 7 0 5 4 6 8) -
-> (1 2 3 7 6 5 4 0 8) --> (1 2 3 7 6 5 4 8 0) --> (1 2 3 7 6 0 4 8 5) --> (1 2 3 7 0 6 4 8 5
) --> (1 2 3 0 7 6 4 8 5) --> (1 2 3 4 7 6 0 8 5) --> (1 2 3 4 7 6 8 0 5) --> (1 2 3 4 7 6 8
5 0) --> (1 2 3 4 7 0 8 5 6) --> (1 2 3 4 0 7 8 5 6) --> (1 2 3 4 5 7 8 0 6) --> (1 2 3 4 5
7 0 8 6) --> (1 2 3 0 5 7 4 8 6) --> (1 2 3 5 0 7 4 8 6) --> (1 2 3 5 7 0 4 8 6) --> (1 2 3
5 7 6 4 8 0) --> (1 2 3 5 7 6 4 0 8) --> (1 2 3 5 0 6 4 7 8) --> (1 2 3 0 5 6 4 7 8) --> (1
2 3 4 5 6 0 7 8) --> (1 2 3 4 5 6 7 0 8) --> (1 2 3 4 5 6 7 8 0)

v. **Initial-state: [[1, 3, 0], [4, 2, 5], [7, 8, 6]] → (1 3 0 4 2 5 7 8 6)**
Puzzle Solved
Number of steps: 4
(1 0 3 4 2 5 7 8 6) --> (1 2 3 4 0 5 7 8 6) --> (1 2 3 4 5 0 7 8 6) --> (1 2 3 4 5 6 7 8 0)

Heuristic 2: Manhattan Distance

i. **Initial-state: [[4, 5, 0], [6, 1, 8], [7, 3, 2]]** ➔ **(4 5 0 6 1 8 7 3 2)**
Could not find solution

ii. **Initial-state: [[1, 2, 5], [0, 4, 3], [6, 8, 7]]** ➔ **(1 2 5 0 4 3 6 8 7)**
Puzzle Solved
Number of steps: 49
(1 2 5 4 0 3 6 8 7) --> (1 2 5 4 8 3 6 0 7) --> (1 2 5 4 8 3 6 7 0) --> (1 2 5 4 8 0 6 7 3) -
-> (1 2 0 4 8 5 6 7 3) --> (1 0 2 4 8 5 6 7 3) --> (1 8 2 4 0 5 6 7 3) --> (1 8 2 4 5 0 6 7 3
) --> (1 8 2 4 5 3 6 7 0) --> (1 8 2 4 5 3 6 0 7) --> (1 8 2 4 0 3 6 5 7) --> (1 0 2 4 8 3 6
5 7) --> (1 2 0 4 8 3 6 5 7) --> (1 2 3 4 8 0 6 5 7) --> (1 2 3 4 0 8 6 5 7) --> (1 2 3 4 5
8 6 0 7) --> (1 2 3 4 5 8 6 7 0) --> (1 2 3 4 5 0 6 7 8) --> (1 2 3 4 0 5 6 7 8) --> (1 2 3
4 7 5 6 0 8) --> (1 2 3 4 7 5 0 6 8) --> (1 2 3 0 7 5 4 6 8) --> (1 2 3 7 0 5 4 6 8) --> (1
2 3 7 5 0 4 6 8) --> (1 2 3 7 5 8 4 6 0) --> (1 2 3 7 5 8 4 0 6) --> (1 2 3 7 0 8 4 5 6) -->
(1 2 3 7 8 0 4 5 6) --> (1 2 3 7 8 6 4 5 0) --> (1 2 3 7 8 6 4 0 5) --> (1 2 3 7 0 6 4 8 5) -
-> (1 2 3 0 7 6 4 8 5) --> (1 2 3 4 7 6 0 8 5) --> (1 2 3 4 7 6 8 0 5) --> (1 2 3 4 7 6 8 5 0
) --> (1 2 3 4 7 0 8 5 6) --> (1 2 3 4 0 7 8 5 6) --> (1 2 3 4 5 7 8 0 6) --> (1 2 3 4 5 7 0
8 6) --> (1 2 3 0 5 7 4 8 6) --> (1 2 3 5 0 7 4 8 6) --> (1 2 3 5 7 0 4 8 6) --> (1 2 3 5 7
6 4 8 0) --> (1 2 3 5 7 6 4 0 8) --> (1 2 3 5 0 6 4 7 8) --> (1 2 3 0 5 6 4 7 8) --> (1 2 3
4 5 6 0 7 8) --> (1 2 3 4 5 6 7 0 8) --> (1 2 3 4 5 6 7 8 0)

iii. **Initial state: [[0, 1, 2], [3, 4, 5], [6, 7, 8]]** ➔ **(0 1 2 3 4 5 6 7 8)**
Puzzle Solved
Number of steps: 30
(1 0 2 3 4 5 6 7 8) --> (1 2 0 3 4 5 6 7 8) --> (1 2 5 3 4 0 6 7 8) --> (1 2 5 3 0 4 6 7 8) -
-> (1 2 5 0 3 4 6 7 8) --> (1 2 5 6 3 4 0 7 8) --> (1 2 5 6 3 4 7 0 8) --> (1 2 5 6 3 4 7 8 0
) --> (1 2 5 6 3 0 7 8 4) --> (1 2 5 6 0 3 7 8 4) --> (1 2 5 0 6 3 7 8 4) --> (0 2 5 1 6 3 7
8 4) --> (2 0 5 1 6 3 7 8 4) --> (2 5 0 1 6 3 7 8 4) --> (2 5 3 1 6 0 7 8 4) --> (2 5 3 1 0
6 7 8 4) --> (2 0 3 1 5 6 7 8 4) --> (0 2 3 1 5 6 7 8 4) --> (1 2 3 0 5 6 7 8 4) --> (1 2 3
7 5 6 0 8 4) --> (1 2 3 7 5 6 8 0 4) --> (1 2 3 7 5 6 8 4 0) --> (1 2 3 7 5 0 8 4 6) --> (1
2 3 7 0 5 8 4 6) --> (1 2 3 7 4 5 8 0 6) --> (1 2 3 7 4 5 0 8 6) --> (1 2 3 0 4 5 7 8 6) -->
(1 2 3 4 0 5 7 8 6) --> (1 2 3 4 5 0 7 8 6) --> (1 2 3 4 5 6 7 8 0)

iv. **Initial-state: [[1, 2, 3], [4, 0, 5], [6, 7, 8]]** ➔ **(1 2 3 4 0 5 6 7 8)**
Puzzle Solved
Number of steps: 30
(1 2 3 4 7 5 6 0 8) --> (1 2 3 4 7 5 0 6 8) --> (1 2 3 0 7 5 4 6 8) --> (1 2 3 7 0 5 4 6 8) -
-> (1 2 3 7 5 0 4 6 8) --> (1 2 3 7 5 8 4 6 0) --> (1 2 3 7 5 8 4 0 6) --> (1 2 3 7 0 8 4 5 6
) --> (1 2 3 7 8 0 4 5 6) --> (1 2 3 7 8 6 4 5 0) --> (1 2 3 7 8 6 4 0 5) --> (1 2 3 7 0 6 4
8 5) --> (1 2 3 0 7 6 4 8 5) --> (1 2 3 4 7 6 0 8 5) --> (1 2 3 4 7 6 8 0 5) --> (1 2 3 4 7
6 8 5 0) --> (1 2 3 4 7 0 8 5 6) --> (1 2 3 4 0 7 8 5 6) --> (1 2 3 4 5 7 8 0 6) --> (1 2 3
4 5 7 0 8 6) --> (1 2 3 0 5 7 4 8 6) --> (1 2 3 5 0 7 4 8 6) --> (1 2 3 5 7 0 4 8 6) --> (1
2 3 5 7 6 4 8 0) --> (1 2 3 5 7 6 4 0 8) --> (1 2 3 5 0 6 4 7 8) --> (1 2 3 0 5 6 4 7 8) -->
(1 2 3 4 5 6 0 7 8) --> (1 2 3 4 5 6 7 0 8) --> (1 2 3 4 5 6 7 8 0)

v. **Initial-state: [[1, 3, 0], [4, 2, 5], [7, 8, 6]] → (1 3 0 4 2 5 7 8 6)**
Puzzle Solved
Number of steps: 4
(1 0 3 4 2 5 7 8 6) --> (1 2 3 4 0 5 7 8 6) --> (1 2 3 4 5 0 7 8 6) --> (1 2 3 4 5 6 7 8 0)

Heuristic 3: Maximum matrix Distance

i. **Initial-state: [[4, 5, 0], [6, 1, 8], [7, 3, 2]] → (4 5 0 6 1 8 7 3 2)**
Could not find solution

ii. **Initial-state: [[1, 2, 5], [0, 4, 3], [6, 8, 7]] → (1 2 5 0 4 3 6 8 7)**
Puzzle Solved
Number of steps: 27
(1 2 5 4 0 3 6 8 7) --> (1 2 5 4 8 3 6 0 7) --> (1 2 5 4 8 3 0 6 7) --> (1 2 5 0 8 3 4 6 7) --
-> (1 2 5 8 0 3 4 6 7) --> (1 2 5 8 6 3 4 0 7) --> (1 2 5 8 6 3 4 7 0) --> (1 2 5 8 6 0 4 7 3
) --> (1 2 5 8 0 6 4 7 3) --> (1 2 5 0 8 6 4 7 3) --> (1 2 5 4 8 6 0 7 3) --> (1 2 5 4 8 6 7
0 3) --> (1 2 5 4 0 6 7 8 3) --> (1 2 5 4 6 0 7 8 3) --> (1 2 0 4 6 5 7 8 3) --> (1 0 2 4 6
5 7 8 3) --> (1 6 2 4 0 5 7 8 3) --> (1 6 2 4 5 0 7 8 3) --> (1 6 2 4 5 3 7 8 0) --> (1 6 2
4 5 3 7 0 8) --> (1 6 2 4 0 3 7 5 8) --> (1 0 2 4 6 3 7 5 8) --> (1 2 0 4 6 3 7 5 8) --> (1
2 3 4 6 0 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)

iii. **Initial state: [[0, 1, 2], [3, 4, 5], [6, 7, 8]] → (0 1 2 3 4 5 6 7 8)**
Puzzle Solved
Number of steps: 38
(1 0 2 3 4 5 6 7 8) --> (1 4 2 3 0 5 6 7 8) --> (1 4 2 0 3 5 6 7 8) --> (1 4 2 6 3 5 0 7 8) --
-> (1 4 2 6 3 5 7 0 8) --> (1 4 2 6 3 5 7 8 0) --> (1 4 2 6 3 0 7 8 5) --> (1 4 2 6 0 3 7 8 5
) --> (1 0 2 6 4 3 7 8 5) --> (1 2 0 6 4 3 7 8 5) --> (1 2 3 6 4 0 7 8 5) --> (1 2 3 6 4 5 7
8 0) --> (1 2 3 6 4 5 7 0 8) --> (1 2 3 6 0 5 7 4 8) --> (1 2 3 0 6 5 7 4 8) --> (1 2 3 7 6
5 0 4 8) --> (1 2 3 7 6 5 4 0 8) --> (1 2 3 7 6 5 4 8 0) --> (1 2 3 7 6 0 4 8 5) --> (1 2 3
7 0 6 4 8 5) --> (1 2 3 0 7 6 4 8 5) --> (1 2 3 4 7 6 0 8 5) --> (1 2 3 4 7 6 8 0 5) --> (1
2 3 4 7 6 8 5 0) --> (1 2 3 4 7 0 8 5 6) --> (1 2 3 4 0 7 8 5 6) --> (1 2 3 4 5 7 8 0 6) -->
(1 2 3 4 5 7 0 8 6) --> (1 2 3 0 5 7 4 8 6) --> (1 2 3 5 0 7 4 8 6) --> (1 2 3 5 7 0 4 8 6) --
-> (1 2 3 5 7 6 4 8 0) --> (1 2 3 5 7 6 4 0 8) --> (1 2 3 5 0 6 4 7 8) --> (1 2 3 0 5 6 4 7 8
) --> (1 2 3 4 5 6 0 7 8) --> (1 2 3 4 5 6 7 0 8) --> (1 2 3 4 5 6 7 8 0)

iv. **Initial-state: [[1, 2, 3], [4, 0, 5], [6, 7, 8]] → (1 2 3 4 0 5 6 7 8)**
Puzzle Solved
Number of steps: 26
(1 2 3 4 7 5 6 0 8) --> (1 2 3 4 7 5 0 6 8) --> (1 2 3 0 7 5 4 6 8) --> (1 2 3 7 0 5 4 6 8) --
-> (1 2 3 7 5 0 4 6 8) --> (1 2 3 7 5 8 4 6 0) --> (1 2 3 7 5 8 4 0 6) --> (1 2 3 7 5 8 0 4 6
) --> (1 2 3 0 5 8 7 4 6) --> (1 2 3 5 0 8 7 4 6) --> (1 2 3 5 8 0 7 4 6) --> (1 2 3 5 8 6 7
4 0) --> (1 2 3 5 8 6 7 0 4) --> (1 2 3 5 0 6 7 8 4) --> (1 2 3 0 5 6 7 8 4) --> (1 2 3 7 5
6 0 8 4) --> (1 2 3 7 5 6 8 0 4) --> (1 2 3 7 5 6 8 4 0) --> (1 2 3 7 5 0 8 4 6) --> (1 2 3
7 0 5 8 4 6) --> (1 2 3 7 4 5 8 0 6) --> (1 2 3 7 4 5 0 8 6) --> (1 2 3 0 4 5 7 8 6) --> (1
2 3 4 0 5 7 8 6) --> (1 2 3 4 5 0 7 8 6) --> (1 2 3 4 5 6 7 8 0)

    v.   **Initial-state: [[1, 3, 0], [4, 2, 5], [7, 8, 6]] → (1 3 0 4 2 5 7 8 6)**
Puzzle Solved
Number of steps: 4
(1 0 3 4 2 5 7 8 6) --> (1 2 3 4 0 5 7 8 6) --> (1 2 3 4 5 0 7 8 6) --> (1 2 3 4 5 6 7 8 0)

Average number of steps:

Heuristic 1 - Number of misplaced Tiles: 21.0

Heuristic 2 - Manhattan Distance: 22.6

Heuristic 3 - Maximum Matrix Distance: 19.0

2. **A* (A-star)**

Heuristic 1: Number of misplaced tiles

    i.   **Initial-state: [[4, 5, 0], [6, 1, 8], [7, 3, 2]] → (4 5 0 6 1 8 7 3 2)**
Could not find solution

    ii.   **Initial-state: [[1, 2, 5], [0, 4, 3], [6, 8, 7]] → (1 2 5 0 4 3 6 8 7)**
Could not find solution.

    iii.   **Initial state: [[0, 1, 2], [3, 4, 5], [6, 7, 8]] → (0 1 2 3 4 5 6 7 8)**
Could not find solution.

    iv.   **Initial-state: [[1, 2, 3], [4, 0, 5], [6, 7, 8]] → (1 2 3 4 0 5 6 7 8)**
Puzzle Solved
Number of steps: 14
(1 2 3 4 5 0 6 7 8) --> (1 2 3 4 5 8 6 7 0) --> (1 2 3 4 5 8 6 0 7) --> (1 2 3 4 5 8 0 6 7) -
-> (1 2 3 0 5 8 4 6 7) --> (1 2 3 5 0 8 4 6 7) --> (1 2 3 5 6 8 4 0 7) --> (1 2 3 5 6 8 4 7 0
) --> (1 2 3 5 6 0 4 7 8) --> (1 2 3 5 0 6 4 7 8) --> (1 2 3 0 5 6 4 7 8) --> (1 2 3 4 5 6 0
7 8) --> (1 2 3 4 5 6 7 0 8) --> (1 2 3 4 5 6 7 8 0)

    v.   **Initial-state: [[1, 3, 0], [4, 2, 5], [7, 8, 6]] → (1 3 0 4 2 5 7 8 6)**
Puzzle Solved
Number of steps: 4
(1 0 3 4 2 5 7 8 6) --> (1 2 3 4 0 5 7 8 6) --> (1 2 3 4 5 0 7 8 6) --> (1 2 3 4 5 6 7 8 0)

Heuristic 2: Manhattan Distance

    i.   **Initial-state: [[4, 5, 0], [6, 1, 8], [7, 3, 2]] → (4 5 0 6 1 8 7 3 2)**
Could not find solution

    ii.   **Initial-state: [[1, 2, 5], [0, 4, 3], [6, 8, 7]] → (1 2 5 0 4 3 6 8 7)**
Could not find solution.

iii. **Initial state: [[0, 1, 2], [3, 4, 5], [6, 7, 8]] → (0 1 2 3 4 5 6 7 8)**
Puzzle Solved
Number of steps: 19
(1 2 5 4 0 3 6 8 7) --> (1 2 5 4 8 3 6 0 7) --> (1 2 5 4 8 3 0 6 7) --> (1 2 5 0 8 3 4 6 7) --> (0 2 5 1 8 3 4 6 7) --> (2 0 5 1 8 3 4 6 7) --> (2 5 0 1 8 3 4 6 7) --> (2 5 3 1 8 0 4 6 7) --> (2 5 3 1 0 8 4 6 7) --> (2 5 3 1 6 8 4 0 7) --> (2 5 3 1 6 8 4 7 0) --> (2 5 3 1 6 0 4 7 8) --> (2 5 3 1 0 6 4 7 8) --> (2 0 3 1 5 6 4 7 8) --> (0 2 3 1 5 6 4 7 8) --> (1 2 3 0 5 6 4 7 8) --> (1 2 3 4 5 6 0 7 8) --> (1 2 3 4 5 6 7 0 8) --> (1 2 3 4 5 6 7 8 0)

iv. **Initial-state: [[1, 2, 3], [4, 0, 5], [6, 7, 8]] → (1 2 3 4 0 5 6 7 8)**
Puzzle Solved
Number of steps: 14
(1 2 3 4 5 0 6 7 8) --> (1 2 3 4 5 8 6 7 0) --> (1 2 3 4 5 8 6 0 7) --> (1 2 3 4 5 8 0 6 7) --> (1 2 3 0 5 8 4 6 7) --> (1 2 3 5 0 8 4 6 7) --> (1 2 3 5 6 8 4 0 7) --> (1 2 3 5 6 8 4 7 0) --> (1 2 3 5 6 0 4 7 8) --> (1 2 3 5 0 6 4 7 8) --> (1 2 3 0 5 6 4 7 8) --> (1 2 3 4 5 6 0 7 8) --> (1 2 3 4 5 6 7 0 8) --> (1 2 3 4 5 6 7 8 0)

v. **Initial-state: [[1, 3, 0], [4, 2, 5], [7, 8, 6]] → (1 3 0 4 2 5 7 8 6)**
Puzzle Solved
Number of steps: 4
(1 0 3 4 2 5 7 8 6) --> (1 2 3 4 0 5 7 8 6) --> (1 2 3 4 5 0 7 8 6) --> (1 2 3 4 5 6 7 8 0)

Heuristic 3: Maximum matrix distance

i. **Initial-state: [[4, 5, 0], [6, 1, 8], [7, 3, 2]] → (4 5 0 6 1 8 7 3 2)**
Could not find solution

ii. **Initial-state: [[1, 2, 5], [0, 4, 3], [6, 8, 7]] → (1 2 5 0 4 3 6 8 7)**
Puzzle Solved
Number of steps: 19
(1 2 5 4 0 3 6 8 7) --> (1 2 5 4 8 3 6 0 7) --> (1 2 5 4 8 3 0 6 7) --> (1 2 5 0 8 3 4 6 7) --> (0 2 5 1 8 3 4 6 7) --> (2 0 5 1 8 3 4 6 7) --> (2 5 0 1 8 3 4 6 7) --> (2 5 3 1 8 0 4 6 7) --> (2 5 3 1 0 8 4 6 7) --> (2 5 3 1 6 8 4 0 7) --> (2 5 3 1 6 8 4 7 0) --> (2 5 3 1 6 0 4 7 8) --> (2 0 3 1 5 6 4 7 8) --> (0 2 3 1 5 6 4 7 8) --> (1 2 3 0 5 6 4 7 8) --> (1 2 3 4 5 6 0 7 8) --> (1 2 3 4 5 6 7 0 8) --> (1 2 3 4 5 6 7 8 0)

iii. **Initial state: [[0, 1, 2], [3, 4, 5], [6, 7, 8]] → (0 1 2 3 4 5 6 7 8)**
Could not find solution.

iv. **Initial-state: [[1, 2, 3], [4, 0, 5], [6, 7, 8]] → (1 2 3 4 0 5 6 7 8)**
Puzzle Solved
Number of steps: 14
(1 2 3 4 5 0 6 7 8) --> (1 2 3 4 5 8 6 7 0) --> (1 2 3 4 5 8 6 0 7) --> (1 2 3 4 5 8 0 6 7) --> (1 2 3 0 5 8 4 6 7) --> (1 2 3 5 0 8 4 6 7) --> (1 2 3 5 6 8 4 0 7) --> (1 2 3 5 6 8 4 7 0) --> (1 2 3 5 6 0 4 7 8) --> (1 2 3 5 0 6 4 7 8) --> (1 2 3 0 5 6 4 7 8) --> (1 2 3 4 5 6 0 7 8) --> (1 2 3 4 5 6 7 0 8) --> (1 2 3 4 5 6 7 8 0)

v. **Initial-state: [[1, 3, 0], [4, 2, 5], [7, 8, 6]] → (1 3 0 4 2 5 7 8 6)**
Puzzle Solved
Number of steps: 4
(1 0 3 4 2 5 7 8 6) --> (1 2 3 4 0 5 7 8 6) --> (1 2 3 4 5 0 7 8 6) --> (1 2 3 4 5 6 7 8 0)

Average number of steps:

Heuristic 1 - Number of misplaced Tiles: 3.6

Heuristic 2 - Manhattan Distance: 11.8

Heuristic 3 - Maximum Matrix Distance: 7.4

Conclusion from result:

From the results, we can see that A*'s average number of steps are less when compared to BFS average steps, for all the heuristics (misplaced tiles, Manhattan distance and Chebyshev distance).
A* combines the advantages of Best-first Search and Uniform Cost Search and it achieves better performance.
For BFS the average number of steps for all three heuristics is greater than 20, in A* the average number of steps are close to 10.

Steps to execute the program:

Input: No need

Output: first it displays the output generated by BFS and then A* output for all three heuristics for all 5 puzzles.

To change the puzzle initial state: In main function (end of file), all the 5 initial state are written in 3x3 matrix format. It can be changed before execution.

To change the goal state: goal state is declared as a global variable. (at the top).

15-Puzzle (extra)

1. **BFS - Best First Search**

    Average number of steps
    Heuristic 1 - Number of misplaced Tiles: 5.4
    Heuristic 2 - Manhattan Distance: 5.4
    Heuristic 3 - Maximum Matrix Distance: 5.4

2. **A* - A-star**

    Average number of steps
    Heuristic 1 - Number of misplaced Tiles: 5.4
    Heuristic 2 - Manhattan Distance: 5.4
    Heuristic 3 - Maximum Matrix Distance: 5.4

Conclusion from results:

Both BFS and A* performed in a similar way, with equal number of average steps.

Steps to execute 15-puzzle program:

This code is written in different file. The execution is similar to 8 puzzle program.

Input: No need

Output: first it displays the output generated by BFS and then A* output for all three heuristics for all 5 puzzles.

To change the puzzle initial state: In main function (end of file), all the 5 initial state are written in 4x4 matrix format. It can be changed before execution.

To change the goal state: goal state is declared as a global variable. (at the top).