****

|  |  |
| --- | --- |
| **Name** | **ID** |
| Khaled Gamal | 6117 |
| Merna Mourad | 6377 |
| Khaled Hesham | 6434 |
| Omar Sherif | 6540 |

**Subject:** Artificial intelligence

**Assignment:** Assignment 1 - Using Informed and Uninformed Search Algorithms to Solve 8-Puzzle

**Description:**

* An instance of the 8-puzzle game consists of a board holding 8 distinct movable tiles, plus an empty space. For any such board, the empty space may be legally swapped with any tile horizontally or vertically next to it. In this assignment, the blank space is going to be represented with the number 0.
* After the initial state is entered, the search problem is to find a sequence of moves that transitions this state to the goal state, that is, the configuration with all tiles arranged in ascending order 0,1,2,3,4,5,6,7,8 .
* The search space is the set of all possible states reachable from the initial state. The blank space may be swapped with a component in one of the four directions `**Up', `Down', `Left', `Right'**, one move at a time.
* It’s required to implement the game solution using different algorithms, **BFS**, **DFS**, **A\*** (Manhattan Distance & Euclidean Distance)

A picture containing text, crossword puzzle

Description automatically generated

**How to run:**

Enter a solvable example as shown (087654321) and choose the algorithm you want to test

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated**All the states the algorithm went through to solve the puzzle along with the cost , depth, nodes explored and running time are shown.**

**Implemented Algorithms:**

**1.BFS:**

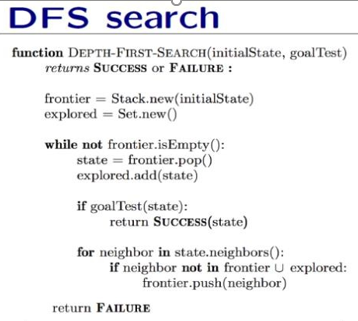
* + The algorithm idea is to horizontally explore the nodes within the tree until we reach the goal.
  + In general its search depth value is fine compared to other algorithms as DFS.
  + The main defect is the long running time because BFS is an uninformed search so we have no information about the path that may help us reach the goal.
  + BFS is implemented using Queue data structure.

Text

Description automatically generated

**2.DFS:**

* + The algorithm idea is to deeply explore the nodes within the tree until we reach the goal.
  + In general its search depth value is bad due to searching until end of depth of each node in tree.
  + Same as BFS the main defect is the long running time because DFS is an uninformed search so we have no information about the path that may help us reach the goal.
  + DFS is implemented using Stack data structure.



**3.A star:**

* + The algorithm idea is to work on finding the goal with putting into consideration the Heuristics that would help reaching the goal in shortest path.
  + A\* is an informed search method, so we have some knowledge that lead us to reach the goal in the shortest and fastest way so its running time is a lot better than any uninformed search method.
  + A\* is implemented using priority queue data structure.
  + For the Heuristics we compute it using any of the following 2 methods **:**

1-Manhattan Distance:

* + - * It is the sum of absolute values of deference’s in the goal's x and y coordinates and the current cell's x and y coordinates respectively.

2-Euclidean Distance:

* + - * It is the distance between the current cell and the goal cell using the distance formula.

A picture containing text

Description automatically generated

**Input sample : 182043765**

**BFS**

States :

1 8 2

0 4 3

7 6 5

---------------------------------------------

1 8 2

4 0 3

7 6 5

---------------------------------------------

1 0 2

4 8 3

7 6 5

---------------------------------------------

0 1 2

4 8 3

7 6 5

---------------------------------------------

4 1 2

0 8 3

7 6 5

---------------------------------------------

4 1 2

7 8 3

0 6 5

---------------------------------------------

4 1 2

7 8 3

6 0 5

---------------------------------------------

4 1 2

7 0 3

6 8 5

---------------------------------------------

4 1 2

7 3 0

6 8 5

---------------------------------------------

4 1 0

7 3 2

6 8 5

---------------------------------------------

4 0 1

7 3 2

6 8 5

---------------------------------------------

4 3 1

7 0 2

6 8 5

---------------------------------------------

4 3 1

0 7 2

6 8 5

---------------------------------------------

0 3 1

4 7 2

6 8 5

---------------------------------------------

3 0 1

4 7 2

6 8 5

---------------------------------------------

3 1 0

4 7 2

6 8 5

---------------------------------------------

3 1 2

4 7 0

6 8 5

---------------------------------------------

3 1 2

4 7 5

6 8 0

---------------------------------------------

3 1 2

4 7 5

6 0 8

---------------------------------------------

3 1 2

4 0 5

6 7 8

---------------------------------------------

3 1 2

0 4 5

6 7 8

---------------------------------------------

0 1 2

3 4 5

6 7 8

---------------------------------------------

Cost = 21

Depth = 22

Nodes Explored = 70958

Running Time = 0.5154092311859131

**DFS**

States:

Last 10 states

3 0 5

4 2 1

6 7 8

---------------------------------------------

3 5 0

4 2 1

6 7 8

---------------------------------------------

3 5 1

4 2 0

6 7 8

---------------------------------------------

3 5 1

4 0 2

6 7 8

---------------------------------------------

3 0 1

4 5 2

6 7 8

---------------------------------------------

3 1 0

4 5 2

6 7 8

---------------------------------------------

3 1 2

4 5 0

6 7 8

---------------------------------------------

3 1 2

4 0 5

6 7 8

---------------------------------------------

3 1 2

0 4 5

6 7 8

---------------------------------------------

0 1 2

3 4 5

6 7 8

---------------------------------------------

Cost = 61183

Depth = 66056

Nodes Explored = 116062

Running Time = 0.45292067527770996

**A Star Search**

1)Manhattan

States :

1 8 2

0 4 3

7 6 5

---------------------------------------------

1 8 2

4 0 3

7 6 5

---------------------------------------------

1 0 2

4 8 3

7 6 5

---------------------------------------------

0 1 2

4 8 3

7 6 5

---------------------------------------------

4 1 2

0 8 3

7 6 5

---------------------------------------------

4 1 2

7 8 3

0 6 5

---------------------------------------------

4 1 2

7 8 3

6 0 5

---------------------------------------------

4 1 2

7 0 3

6 8 5

---------------------------------------------

4 1 2

7 3 0

6 8 5

---------------------------------------------

4 1 0

7 3 2

6 8 5

---------------------------------------------

4 0 1

7 3 2

6 8 5

---------------------------------------------

4 3 1

7 0 2

6 8 5

---------------------------------------------

4 3 1

0 7 2

6 8 5

---------------------------------------------

0 3 1

4 7 2

6 8 5

---------------------------------------------

3 0 1

4 7 2

6 8 5

---------------------------------------------

3 1 0

4 7 2

6 8 5

---------------------------------------------

3 1 2

4 7 0

6 8 5

---------------------------------------------

3 1 2

4 7 5

6 8 0

---------------------------------------------

3 1 2

4 7 5

6 0 8

---------------------------------------------

3 1 2

4 0 5

6 7 8

---------------------------------------------

3 1 2

0 4 5

6 7 8

---------------------------------------------

0 1 2

3 4 5

6 7 8

---------------------------------------------

Cost = 21

Depth = 23

Nodes Explored = 7424

Running Time = 1.5308890342712402

2)Euclidean :

1 8 2

0 4 3

7 6 5

---------------------------------------------

1 8 2

4 0 3

7 6 5

---------------------------------------------

1 0 2

4 8 3

7 6 5

---------------------------------------------

0 1 2

4 8 3

7 6 5

---------------------------------------------

4 1 2

0 8 3

7 6 5

---------------------------------------------

4 1 2

7 8 3

0 6 5

---------------------------------------------

4 1 2

7 8 3

6 0 5

---------------------------------------------

4 1 2

7 0 3

6 8 5

---------------------------------------------

4 1 2

7 3 0

6 8 5

---------------------------------------------

4 1 0

7 3 2

6 8 5

---------------------------------------------

4 0 1

7 3 2

6 8 5

---------------------------------------------

4 3 1

7 0 2

6 8 5

---------------------------------------------

4 3 1

0 7 2

6 8 5

---------------------------------------------

0 3 1

4 7 2

6 8 5

---------------------------------------------

3 0 1

4 7 2

6 8 5

---------------------------------------------

3 1 0

4 7 2

6 8 5

---------------------------------------------

3 1 2

4 7 0

6 8 5

---------------------------------------------

3 1 2

4 7 5

6 8 0

---------------------------------------------

3 1 2

4 7 5

6 0 8

---------------------------------------------

3 1 2

4 0 5

6 7 8

---------------------------------------------

3 1 2

0 4 5

6 7 8

---------------------------------------------

0 1 2

3 4 5

6 7 8

---------------------------------------------

Cost = 21

Depth = 23

Nodes Explored = 7424

Running Time = 1.608999490737915

**Conclusion:**

For the test case: 182043765

|  |  |  |  |
| --- | --- | --- | --- |
| **Search algorithm** | **Cost of path** | **Nodes expanded** | **Running time/sec** |
| BFS | 21 | 70958 | 0.5154 |
| DFS | 61183 | 116062 | 0.4529 |
| A\* Manhattan | 21 | 7424 | 1.5309 |
| A\* Euclidean | 21 | 7424 | 1.609 |

* The cost path is close in all algorithms except for DFS it has very large depth value
* In both uninformed search algorithms (BFS, DFS), the   
  number of expanded nodes is too large, while for informed search algorithm (A\*) the nodes expanded is much smaller, and that’s why Heuristics are powerful.
* The running time truly shows how informed search is much better than uninformed search, as the time difference between BFS,DFS (that are close in running time), and A\* search is great, so that’s why A\* is the best algorithm to solve this game as it expand least nodes and reach the goal in the fastest way.
* concerning the different Heuristics calculating ways, in most cases Manhattan is faster to reach goal

**Test Cases:**

Case 1: 087654321

A\* search euclidean

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A\* search manhanttan

A screenshot of a computer

Description automatically generated

BFS

A screenshot of a computer

Description automatically generated

Case 2 :312045678

BFS

A screenshot of a computer

Description automatically generated

DFS

A screenshot of a computer

Description automatically generated

A\* Manhattan

A screenshot of a computer

Description automatically generated

Euclidean

A screenshot of a computer

Description automatically generated

Case 3:125340678

BFS

A screenshot of a computer

Description automatically generated

DFS

A screenshot of a computer

Description automatically generated

A\* Manhattan

A screenshot of a computer

Description automatically generated

Euclidean

A screenshot of a computer

Description automatically generated