Introduction to Artificial Intelligence
Assignment 1

Submitted to

Dr. Ayman Khalafallah

Eng. Mohamed Tarek El Habiby

Name	ID
البدري محمد السيد البدري	20010329
باتريك جورج كرومل	20010383
محمد حسن يوسف حسن الطوبجي	20011544
مصطفي خالد كمال احمد	20011923

- ## Data Structures:
  - ❖ Queue
  - ❖ Deque
  - ❖ priority queue
  - ❖ Set
  - ❖ map(dictionary).
- ## Algorithms:
  - ❖ BFS
  - ❖ DFS
  - ❖ A*
- ## Test Cases:

1. Case 1: '123456780'

	BFS	DFS	A* Manhattan	A* Euclidean
Running time(ms)	424	328	15	219
Cost	22	63762	22	22
Nodes Expanded	84195	113906	769	22221
Search Depth	23	67100	22	23

2. Case 2: '876543210'

	BFS	DFS	A* Manhattan	A* Euclidean
Running time(ms)	876	289	3.5	1191
Cost	28	65676	28	28

Nodes Expanded	176853	108205	180	120050
Search Depth	29	67100	28	28

3. Case 3: ‘876345012’

	BFS	DFS	A* Manhattan	A* Euclidean
Running time(ms)	863	171	87	1474
Cost	30	53090	30	30
Nodes Expanded	181263	60873	6342	156093
Search Depth	30	53090	30	30

4. Case 4: ‘512340678’

	BFS	DFS	A* Manhattan	A* Euclidean
Running time(ms)	7	237	1	2
Cost	11	63943	11	11
Nodes Expanded	1048	82745	33	248
Search Depth	22	63943	11	11

5. Case 5: ‘432105678’

	BFS	DFS	A* Manhattan	A* Euclidean
Running time(ms)	1	207	3.5	1191

Cost	6	59290	28	28
Nodes Expanded	72	71601	180	120050
Search Depth	7	59290	28	28

- ## Time analysis:
  - ### Observation:
    - BFS Average Time = 434.2 ms
    - DFS Average Time = 246.4 ms
    - A* Euclidean Average Time = 22 ms
    - A* Manhattan Average Time = 815 ms
  - ### Conclusion:
    - BFS and DFS are almost near in running time, sometimes DFS is faster than BFS because of the huge number of nodes expanded in BFS in some cases.
    - A* Manhattan heuristic is more admissible in the current game than A*Euclidean heuristic which contributes to the faster running time and the smaller number of expanded nodes.
    - Why is the Manhattan heuristic is more admissible than the Euclidean heuristic ?

      The Manhattan distance heuristic calculates the sum of the distances (number of moves) between each tile's current position and its goal position, considering only horizontal and vertical movements. It does not take into account diagonal movements.

      On the other hand, the Euclidean distance heuristic calculates the straight-line distance (hypotenuse) between each tile's current position and its goal position, considering both horizontal and vertical movements. It takes into account diagonal movements and can provide a more accurate estimate of the actual distance.

While the Euclidean distance heuristic may appear to be more accurate, it is not guaranteed to be admissible for the 8-puzzle game. This is because the Euclidean distance can underestimate the number of moves required to reach the goal state. It is possible for tiles to be arranged in such a way that the Euclidean distance suggests a shorter distance than what is actually required due to the constraints of the puzzle.

In contrast, the Manhattan distance heuristic always provides a lower bound on the number of moves required. It is admissible because it considers only the horizontal and vertical movements and ignores any potential shortcuts or diagonal movements. Therefore, it guarantees that the estimated cost will never exceed the actual cost to reach the goal state.