AI

Assignment 1

Names:

Ans Gomaa Elnagar (18010421)

Seif Mohamed Mahmoud Gneedy (18010834)

Abdelrahman Ibrahem Fawzy (18010893)

Abdelrahman Abdelfattah Kassem (18010948)

Problem statement:

Solving the 8-puzzle game using BFS, DFS, and A* with the heuristic function as euclidean distance or manhattan distance.

Assumptions:

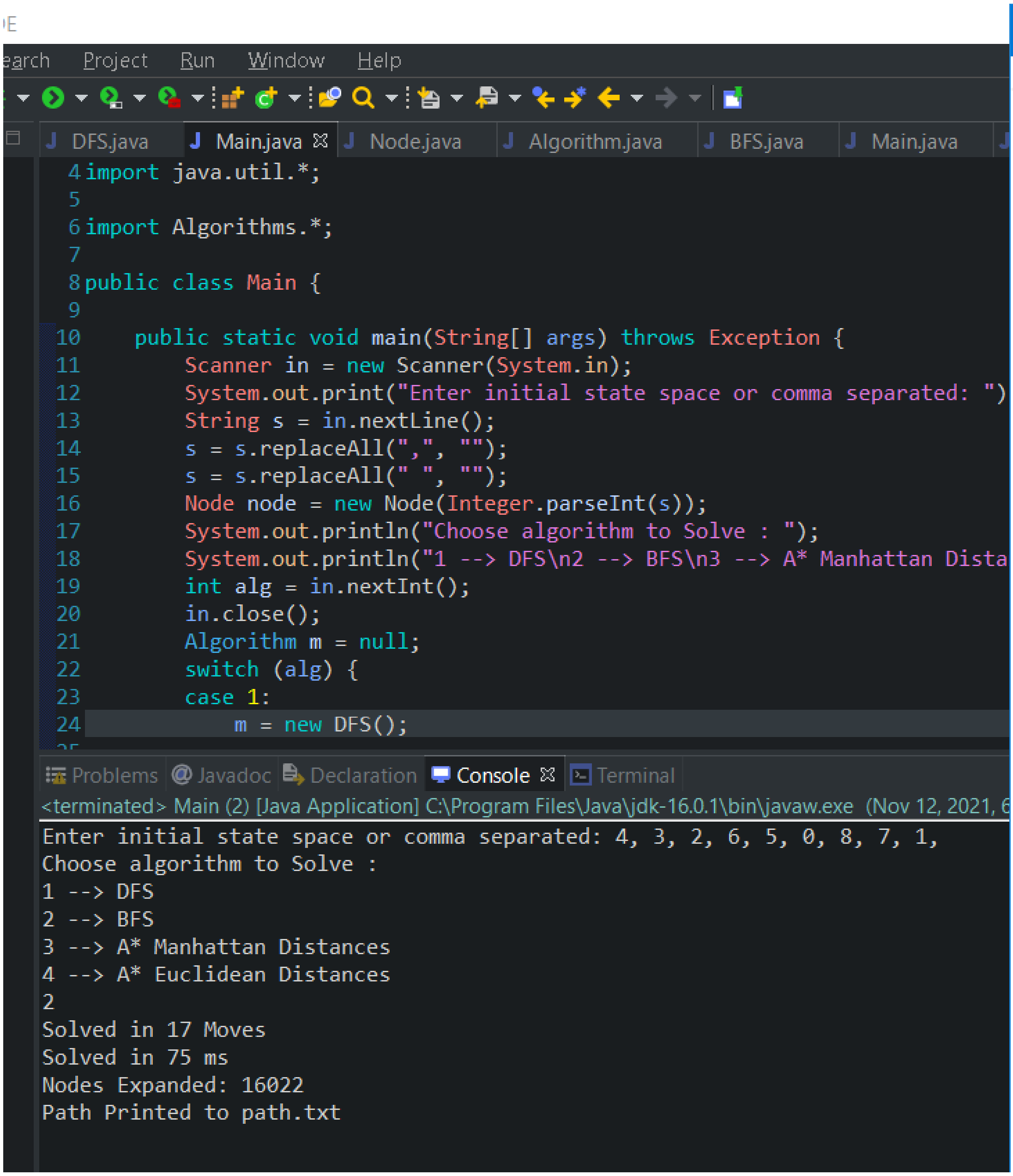
- Code detects invalid and unsolvable states and doesn’t attempt to solve them.
- Enough memory.
- Enough processing power to run in a reasonable time.

Data structures:

- Queue:
  Used in BFS as the frontier list.
- Stack:
  Used in DFS as the frontier list.
- PriorityQueue:
  Used in A* as the frontier list.
- HashSet
  Used for the explored list in all algorithms.
  Also used for the frontier list in BFS and DFS to avoid O(n) running time of Queue and Stack (.contains) methods.
- HashMap
  Used additionally in A* to store the lowest cost for the states that are currently in the frontier only, To avoid O(n) complexity for just checking the lowest cost for a given state in the priority queue(frontier).
- LinkedList
  Used in storing neighbors and in storing the path to the goal.

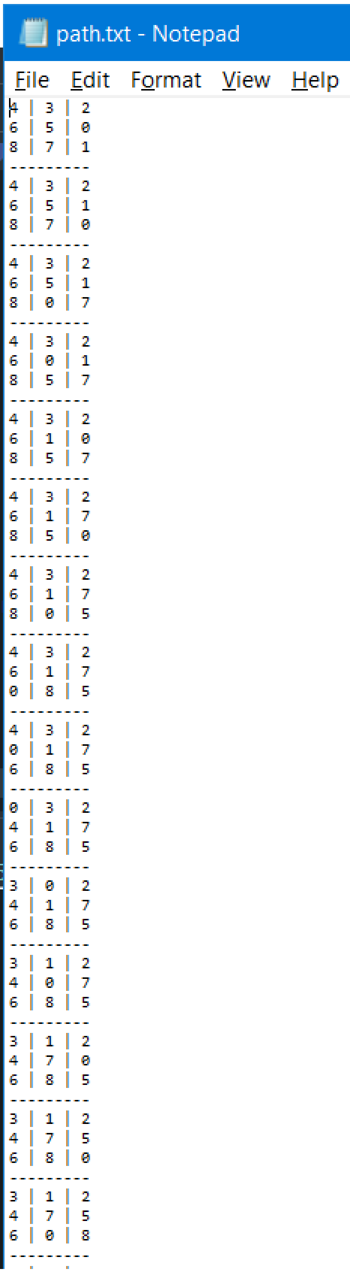
Sample runs:

Run 1:



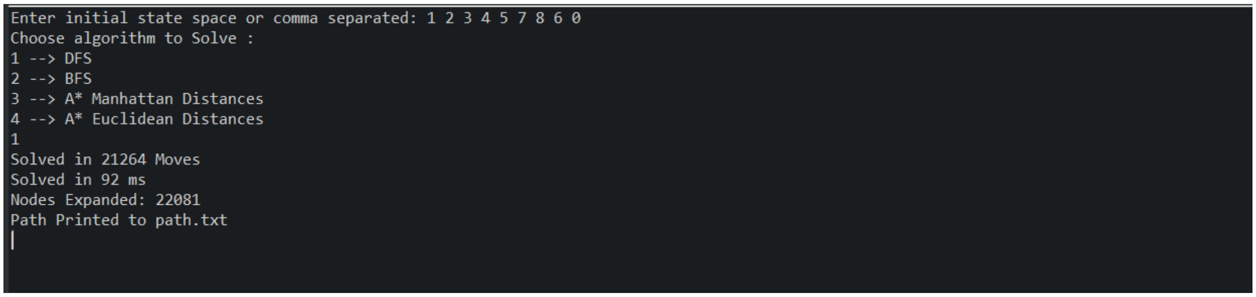
```java
import java.util.*;

import Algorithms.*;

public class Main {

    public static void main(String[] args) throws Exception {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter initial state space or comma separated: ")
        String s = in.nextLine();
        s = s.replaceAll(",", "");
        s = s.replaceAll(" ", "");
        Node node = new Node(Integer.parseInt(s));
        System.out.println("Choose algorithm to Solve : ");
        System.out.println("1 --> DFS\n2 --> BFS\n3 --> A* Manhattan Dista
        int alg = in.nextInt();
        in.close();
        Algorithm m = null;
        switch (alg) {
        case 1:
            m = new DFS();
```

```
<terminated> Main (2) [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (Nov 12, 2021, 6
Enter initial state space or comma separated: 4, 3, 2, 6, 5, 0, 8, 7, 1,
Choose algorithm to Solve :
1 --> DFS
2 --> BFS
3 --> A* Manhattan Distances
4 --> A* Euclidean Distances
2
Solved in 17 Moves
Solved in 75 ms
Nodes Expanded: 16022
Path Printed to path.txt
```



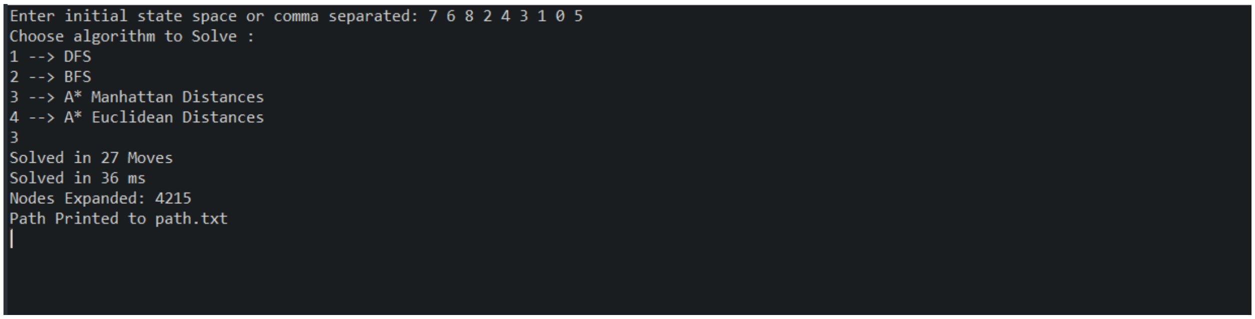
```
4 | 3 | 2
6 | 5 | 0
8 | 7 | 1
----------
4 | 3 | 2
6 | 5 | 1
8 | 7 | 0
----------
4 | 3 | 2
6 | 5 | 1
8 | 0 | 7
----------
4 | 3 | 2
6 | 0 | 1
8 | 5 | 7
----------
4 | 3 | 2
6 | 1 | 0
8 | 5 | 7
----------
4 | 3 | 2
6 | 1 | 7
8 | 5 | 0
----------
4 | 3 | 2
6 | 1 | 7
8 | 0 | 5
----------
4 | 3 | 2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

Run 2:



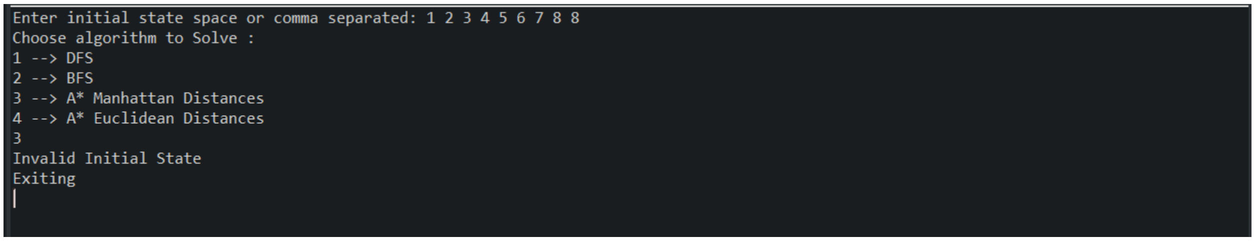
```
Enter initial state space or comma separated: 1 2 3 4 5 7 8 6 0
Choose algorithm to Solve :
1 --> DFS
2 --> BFS
3 --> A* Manhattan Distances
4 --> A* Euclidean Distances
1
Solved in 21264 Moves
Solved in 92 ms
Nodes Expanded: 22081
Path Printed to path.txt
```

Run 3:



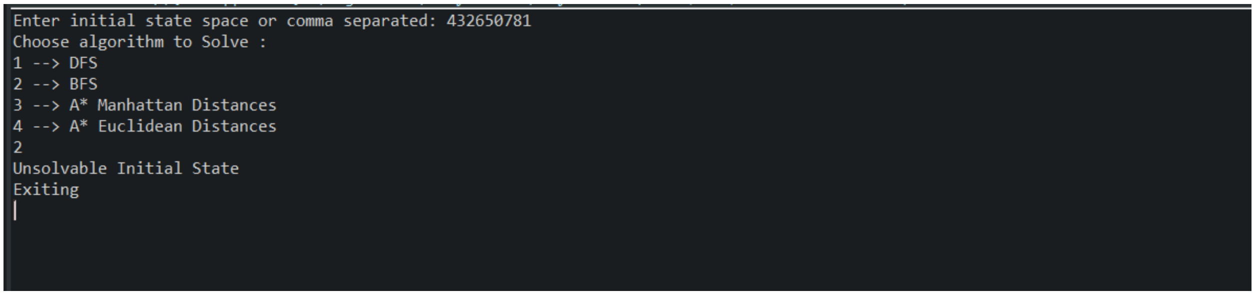
```
Enter initial state space or comma separated: 7 6 8 2 4 3 1 0 5
Choose algorithm to Solve :
1 --> DFS
2 --> BFS
3 --> A* Manhattan Distances
4 --> A* Euclidean Distances
3
Solved in 27 Moves
Solved in 36 ms
Nodes Expanded: 4215
Path Printed to path.txt
```

Run 4:



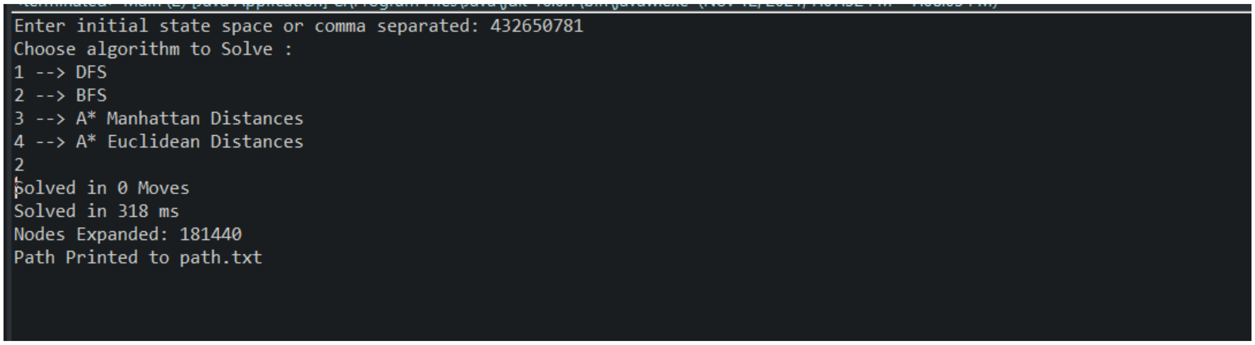
```
Enter initial state space or comma separated: 1 2 3 4 5 6 7 8 8
Choose algorithm to Solve :
1 --> DFS
2 --> BFS
3 --> A* Manhattan Distances
4 --> A* Euclidean Distances
3
Invalid Initial State
Exiting
```

Run 5:



```
Enter initial state space or comma separated: 432650781
Choose algorithm to Solve :
1 --> DFS
2 --> BFS
3 --> A* Manhattan Distances
4 --> A* Euclidean Distances
2
Unsolvable Initial State
Exiting
```

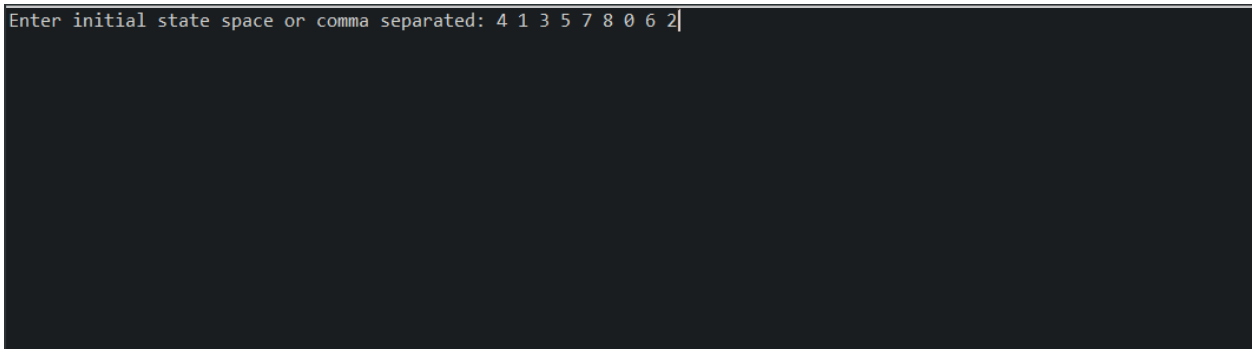
Worst case running time (solvability check is off)



```
Enter initial state space or comma separated: 432650781
Choose algorithm to Solve :
1 --> DFS
2 --> BFS
3 --> A* Manhattan Distances
4 --> A* Euclidean Distances
2
Solved in 0 Moves
Solved in 318 ms
Nodes Expanded: 181440
Path Printed to path.txt
```

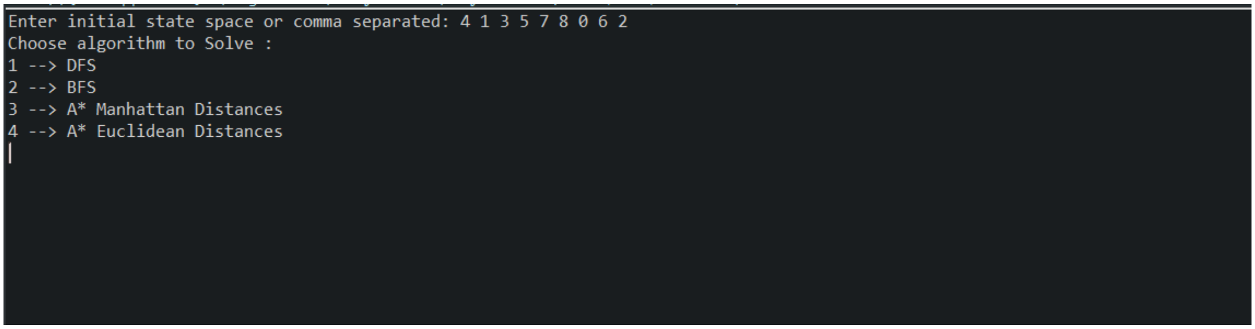
How to use:

Enter initial state



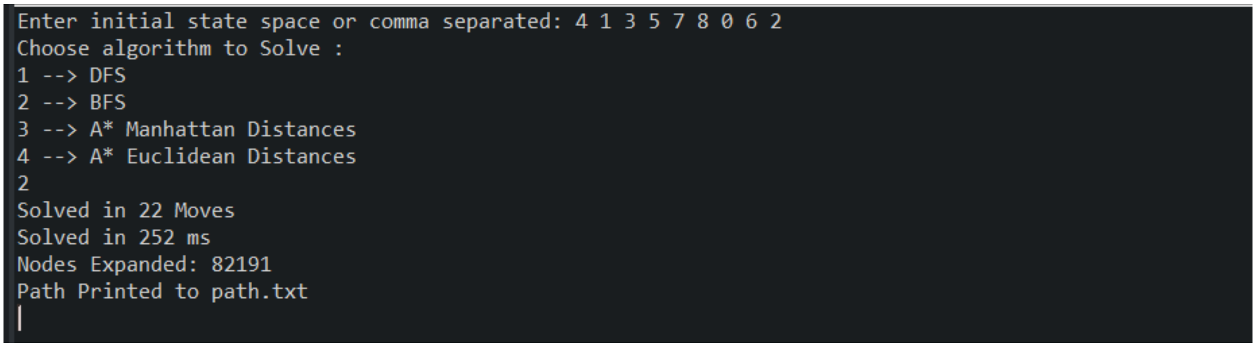
```
Enter initial state space or comma separated: 4 1 3 5 7 8 0 6 2
```

Choose which algorithm to run



```
Enter initial state space or comma separated: 4 1 3 5 7 8 0 6 2
Choose algorithm to Solve :
1 --> DFS
2 --> BFS
3 --> A* Manhattan Distances
4 --> A* Euclidean Distances
```

The detailed path to the goal is saved in the file path.txt



```
Enter initial state space or comma separated: 4 1 3 5 7 8 0 6 2
Choose algorithm to Solve :
1 --> DFS
2 --> BFS
3 --> A* Manhattan Distances
4 --> A* Euclidean Distances
2
Solved in 22 Moves
Solved in 252 ms
Nodes Expanded: 82191
Path Printed to path.txt
```