Search Algorithms to solve 8-puzzle

BFS - DFS - A*

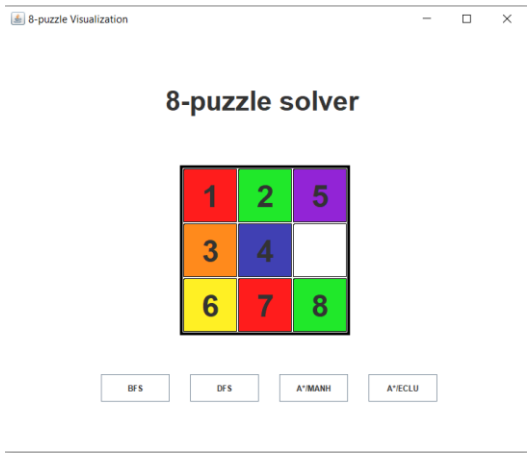
Wadie Bishoy	5074
Nour Shobier	4624
Anas Hamed	4989

Data structures used:

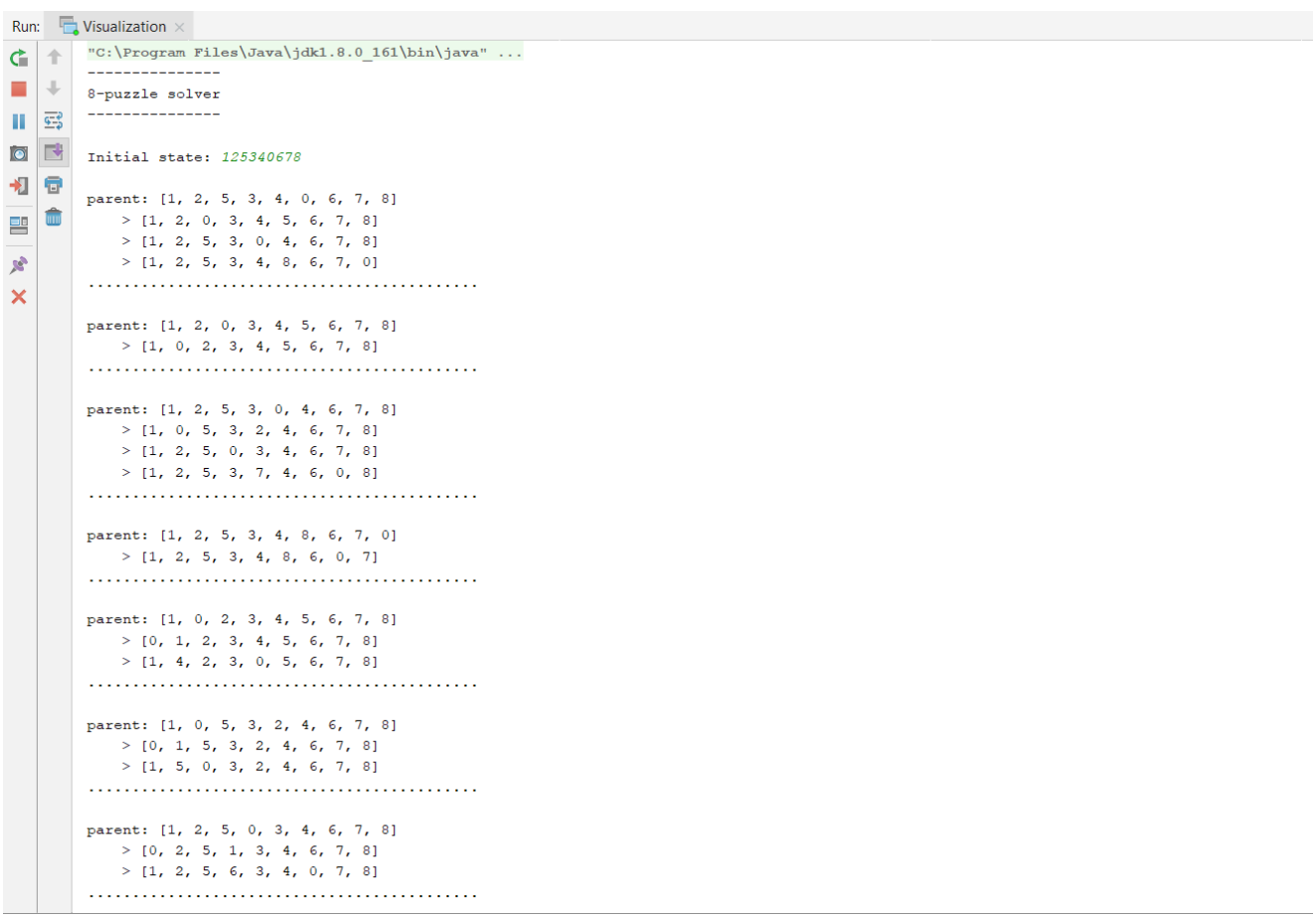
- Hash Maps
  - "explored" hash map is used to keep track of all the nodes were their children were extended (used in both DFS and BFS).
  - "inQueue" hash map is used to keep track of the current nodes in queue (used in BFS).
  - "inStack" hash map is used to keep track of the current nodes in stack (used in DFS).
- Queue
  - "queue" is used in BFS to guarantee that all nodes are explored in each level before it moving to the next level.
- Stack
  - "stack" is used in DFS to guarantee that nodes are explored deeply (not level by level).

1. BFS

- Initial state: 125340678



- Nodes expanded:

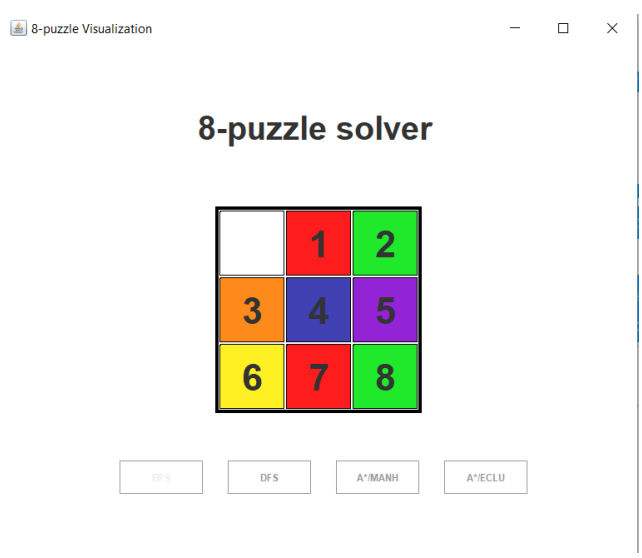


```
Run: Visualization
"C:\Program Files\Java\jdk1.8.0_161\bin\java" ...
---------------
8-puzzle solver
---------------

Initial state: 125340678

parent: [1, 2, 5, 3, 4, 0, 6, 7, 8]
    > [1, 2, 0, 3, 4, 5, 6, 7, 8]
    > [1, 2, 5, 3, 0, 4, 6, 7, 8]
    > [1, 2, 5, 3, 4, 8, 6, 7, 0]
.........................................

parent: [1, 2, 0, 3, 4, 5, 6, 7, 8]
    > [1, 0, 2, 3, 4, 5, 6, 7, 8]
.........................................

parent: [1, 2, 5, 3, 0, 4, 6, 7, 8]
    > [1, 0, 5, 3, 2, 4, 6, 7, 8]
    > [1, 2, 5, 0, 3, 4, 6, 7, 8]
    > [1, 2, 5, 3, 7, 4, 6, 0, 8]
.........................................

parent: [1, 2, 5, 3, 4, 8, 6, 7, 0]
    > [1, 2, 5, 3, 4, 8, 6, 0, 7]
.........................................

parent: [1, 0, 2, 3, 4, 5, 6, 7, 8]
    > [0, 1, 2, 3, 4, 5, 6, 7, 8]
    > [1, 4, 2, 3, 0, 5, 6, 7, 8]
.........................................

parent: [1, 0, 5, 3, 2, 4, 6, 7, 8]
    > [0, 1, 5, 3, 2, 4, 6, 7, 8]
    > [1, 5, 0, 3, 2, 4, 6, 7, 8]
.........................................

parent: [1, 2, 5, 0, 3, 4, 6, 7, 8]
    > [0, 2, 5, 1, 3, 4, 6, 7, 8]
    > [1, 2, 5, 6, 3, 4, 0, 7, 8]
.........................................
```

- path:

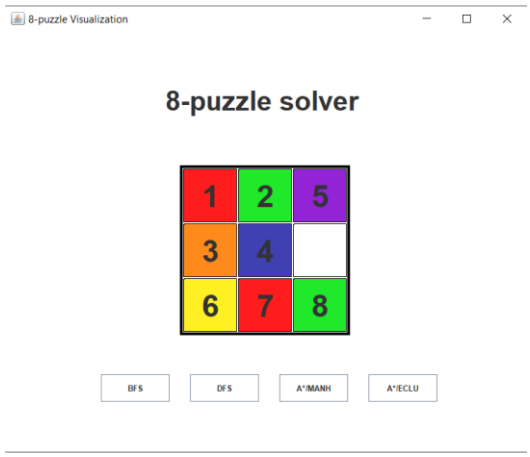
```
parent: [1, 2, 5, 3, 7, 4, 6, 0, 8]
    > [1, 2, 5, 3, 7, 4, 0, 6, 8]
    > [1, 2, 5, 3, 7, 4, 6, 8, 0]
.............................................

parent: [1, 2, 5, 3, 4, 8, 6, 0, 7]
    > [1, 2, 5, 3, 0, 8, 6, 4, 7]
    > [1, 2, 5, 3, 4, 8, 0, 6, 7]
.............................................

parent: [0, 1, 2, 3, 4, 5, 6, 7, 8]
Goal: [0, 1, 2, 3, 4, 5, 6, 7, 8]
Success


no of explored nodes: 10

> Reversed Path:
[0, 1, 2, 3, 4, 5, 6, 7, 8]
[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]
```

- Goal state:

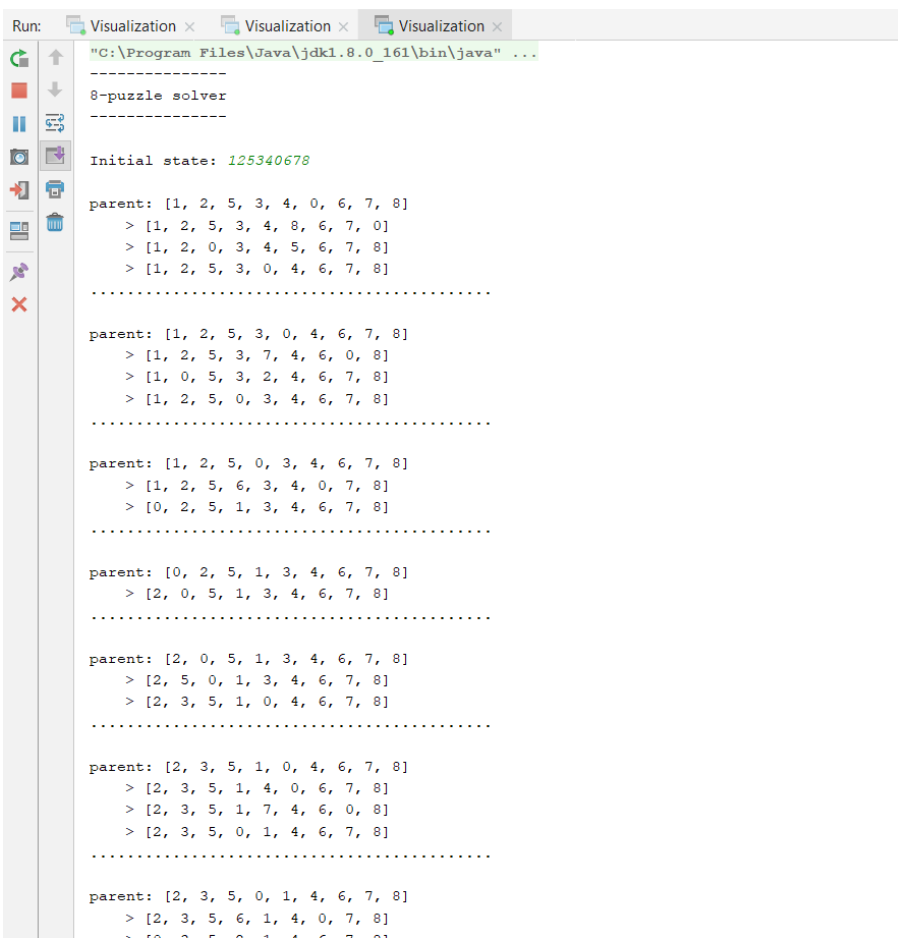


2. DFS

- Initial state: 125340678



- Nodes expanded:

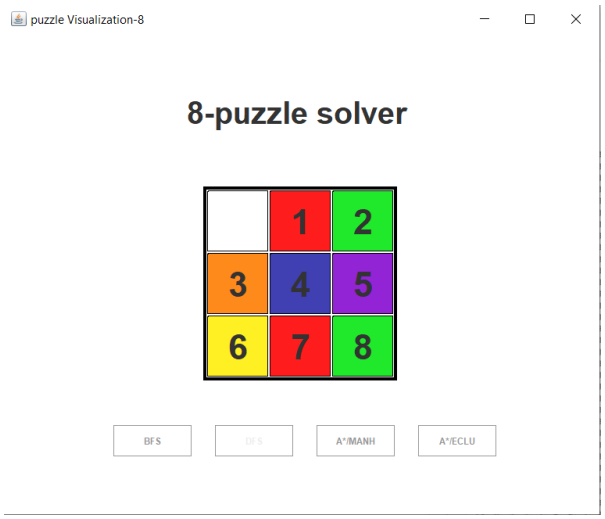


```
Run: Visualization  Visualization  Visualization
"C:\Program Files\Java\jdk1.8.0_161\bin\java" ...
---------------
8-puzzle solver
---------------

Initial state: 125340678

parent: [1, 2, 5, 3, 4, 0, 6, 7, 8]
    > [1, 2, 5, 3, 4, 8, 6, 7, 0]
    > [1, 2, 0, 3, 4, 5, 6, 7, 8]
    > [1, 2, 5, 3, 0, 4, 6, 7, 8]
.........................................

parent: [1, 2, 5, 3, 0, 4, 6, 7, 8]
    > [1, 2, 5, 3, 7, 4, 6, 0, 8]
    > [1, 0, 5, 3, 2, 4, 6, 7, 8]
    > [1, 2, 5, 0, 3, 4, 6, 7, 8]
.........................................

parent: [1, 2, 5, 0, 3, 4, 6, 7, 8]
    > [1, 2, 5, 6, 3, 4, 0, 7, 8]
    > [0, 2, 5, 1, 3, 4, 6, 7, 8]
.........................................

parent: [0, 2, 5, 1, 3, 4, 6, 7, 8]
    > [2, 0, 5, 1, 3, 4, 6, 7, 8]
.........................................

parent: [2, 0, 5, 1, 3, 4, 6, 7, 8]
    > [2, 5, 0, 1, 3, 4, 6, 7, 8]
    > [2, 3, 5, 1, 0, 4, 6, 7, 8]
.........................................

parent: [2, 3, 5, 1, 0, 4, 6, 7, 8]
    > [2, 3, 5, 1, 4, 0, 6, 7, 8]
    > [2, 3, 5, 1, 7, 4, 6, 0, 8]
    > [2, 3, 5, 0, 1, 4, 6, 7, 8]
.........................................

parent: [2, 3, 5, 0, 1, 4, 6, 7, 8]
    > [2, 3, 5, 6, 1, 4, 0, 7, 8]
```

- path:

```
Run:  Visualization  Visualization  Visualization
> Reversed Path:
[0, 1, 2, 3, 4, 5, 6, 7, 8]
[3, 1, 2, 0, 4, 5, 6, 7, 8]
[3, 1, 2, 4, 0, 5, 6, 7, 8]
[3, 0, 2, 4, 1, 5, 6, 7, 8]
[0, 3, 2, 4, 1, 5, 6, 7, 8]
[4, 3, 2, 0, 1, 5, 6, 7, 8]
[4, 3, 2, 1, 0, 5, 6, 7, 8]
[4, 0, 2, 1, 3, 5, 6, 7, 8]
[0, 4, 2, 1, 3, 5, 6, 7, 8]
[1, 4, 2, 0, 3, 5, 6, 7, 8]
[1, 4, 2, 3, 0, 5, 6, 7, 8]
[1, 4, 2, 3, 5, 0, 6, 7, 8]
[1, 4, 0, 3, 5, 2, 6, 7, 8]
[1, 0, 4, 3, 5, 2, 6, 7, 8]
[0, 1, 4, 3, 5, 2, 6, 7, 8]
[3, 1, 4, 0, 5, 2, 6, 7, 8]
[3, 1, 4, 5, 0, 2, 6, 7, 8]
[3, 0, 4, 5, 1, 2, 6, 7, 8]
[0, 3, 4, 5, 1, 2, 6, 7, 8]
[5, 3, 4, 0, 1, 2, 6, 7, 8]
[5, 3, 4, 1, 0, 2, 6, 7, 8]
[5, 0, 4, 1, 3, 2, 6, 7, 8]
[0, 5, 4, 1, 3, 2, 6, 7, 8]
[1, 5, 4, 0, 3, 2, 6, 7, 8]
[1, 5, 4, 3, 0, 2, 6, 7, 8]
[1, 5, 4, 3, 2, 0, 6, 7, 8]
[1, 5, 0, 3, 2, 4, 6, 7, 8]
[1, 0, 5, 3, 2, 4, 6, 7, 8]
[0, 1, 5, 3, 2, 4, 6, 7, 8]
[3, 1, 5, 0, 2, 4, 6, 7, 8]
[3, 1, 5, 2, 0, 4, 6, 7, 8]
[3, 0, 5, 2, 1, 4, 6, 7, 8]
[0, 3, 5, 2, 1, 4, 6, 7, 8]
[2, 3, 5, 0, 1, 4, 6, 7, 8]
[2, 3, 5, 1, 0, 4, 6, 7, 8]
[2, 0, 5, 1, 3, 4, 6, 7, 8]
[0, 2, 5, 1, 3, 4, 6, 7, 8]
[1, 2, 5, 0, 3, 4, 6, 7, 8]
[1, 2, 5, 3, 0, 4, 6, 7, 8]
[1, 2, 5, 3, 4, 0, 6, 7, 8]
```

- Goal state:


puzzle Visualization-8
8-puzzle solver
1
2
3
4
5
6
7
8
BFS
DFS
A*/MANH
A*/ECLU

3. A*

- Solution:

```
Run:  Main ×  Visualization ×  Visualization ×  Main ×
"C:\Program Files\Java\jdk1.8.0_161\bin\java" ...
[1, 2, 3, 0, 4, 6, 7, 5, 8]

Solution is:

1 2 3
0 4 6
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

------------------------
-> Running time: 0ms
-> Cost of path: 3 steps
```