8-puzzle game

Overview:

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 adjacent to it. In this assignment, the blank space is going to be represented with the number 0.

Objective:

Given an initial state of the board, the search problem is to find a sequence of moves that transitions this state to the goal state by applying three algorithms : BFS , DFS and A*;

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. The cost of moving from one configuration of the board to another is the same and equal to one. Thus, the total cost of path is equal to the number of moves made from the initial state to the goal state.

Algorithms:

BFS search

```
function BREADTH-FIRST-SEARCH(initialState, goalTest)
    returns SUCCESS or FAILURE :

    frontier = Queue.new(initialState)
    explored = Set.new()

    while not frontier.isEmpty():
        state = frontier.dequeue()
        explored.add(state)

        if goalTest(state):
            return SUCCESS(state)

        for neighbor in state.neighbors():
            if neighbor not in frontier ∪ explored:
                frontier.enqueue(neighbor)

 return FAILURE
```

DFS search

```
function DEPTH-FIRST-SEARCH(initialState, goalTest)
    returns SUCCESS or FAILURE :

    frontier = Stack.new(initialState)
    explored = Set.new()

    while not frontier.isEmpty():
        state = frontier.pop()
        explored.add(state)

        if goalTest(state):
            return SUCCESS(state)

        for neighbor in state.neighbors():
            if neighbor not in frontier ∪ explored:
                frontier.push(neighbor)

    return FAILURE
```

Taken from the edX course ColumbiaX: CSMM.101x Artificial Intelligence (AI)

A* search

Taken from the edX course ColumbiaX: CSMM.101x Artificial Intelligence (AI)

```
function A-STAR-SEARCH(initialState, goalTest)
    returns SUCCESS or FAILURE : /* Cost f(n) = g(n) + h(n) */

    frontier = Heap.new(initialState)
    explored = Set.new()

    while not frontier.isEmpty():
        state = frontier.deleteMin()
        explored.add(state)

        if goalTest(state):
            return SUCCESS(state)

        for neighbor in state.neighbors():
            if neighbor not in frontier ∪ explored:
                frontier.insert(neighbor)
            else if neighbor in frontier:
                frontier.decreaseKey(neighbor)

    return FAILURE
```

Data Structures Used:

Queue

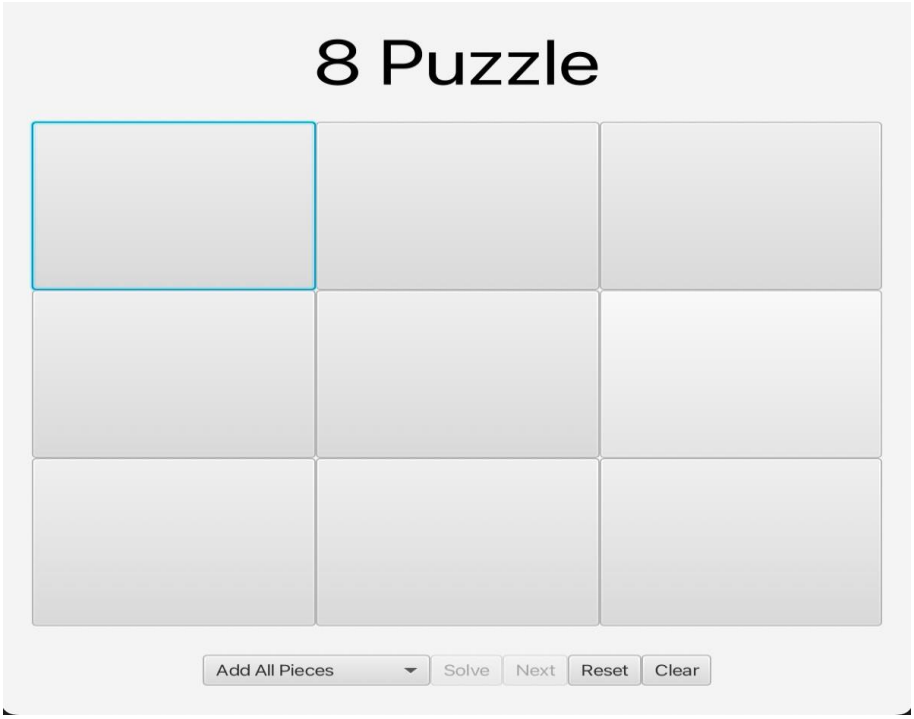
Priority queue

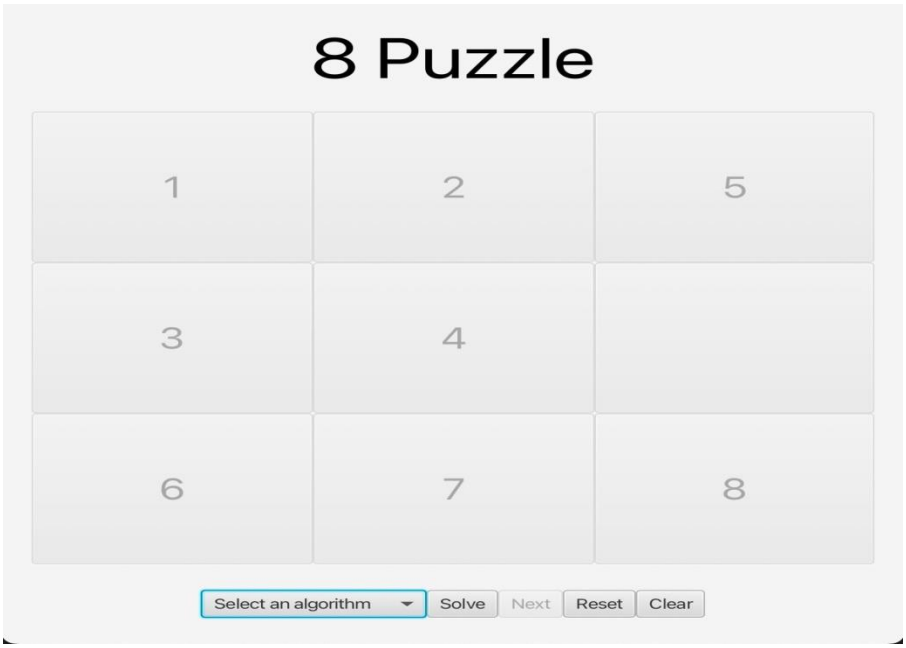
Hash map

Stack

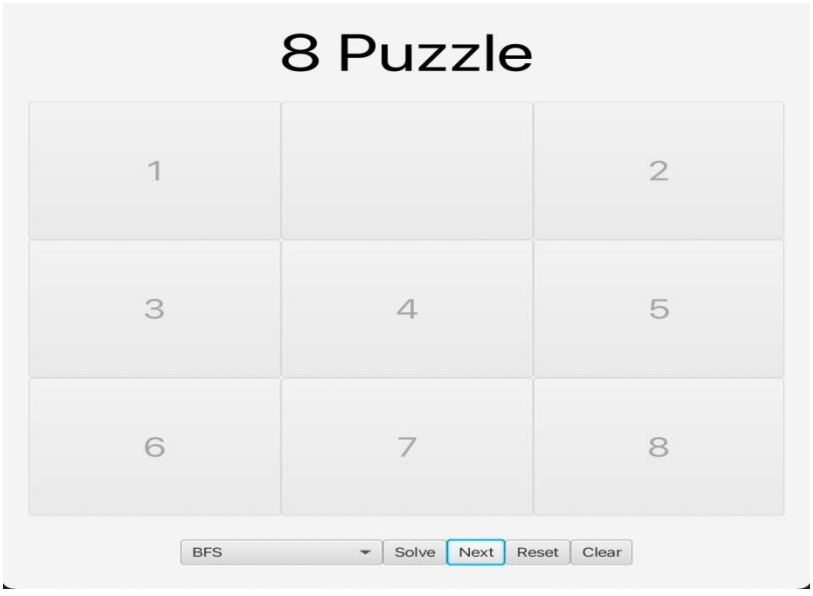
ArrayList

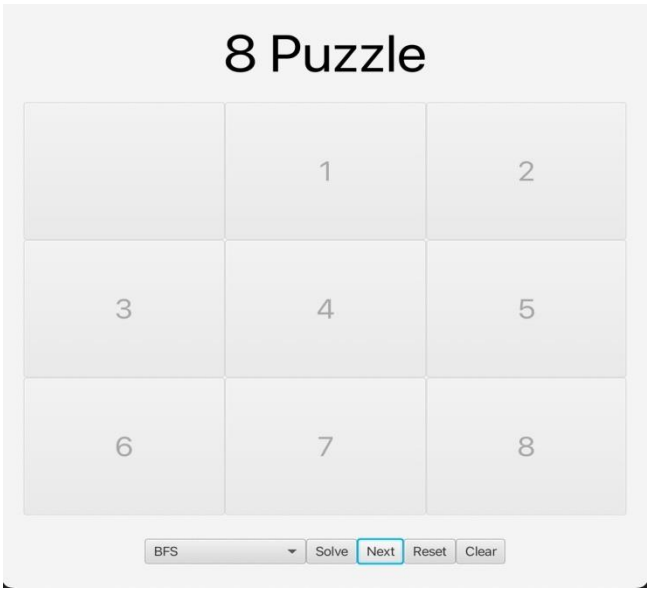
Test Cases:



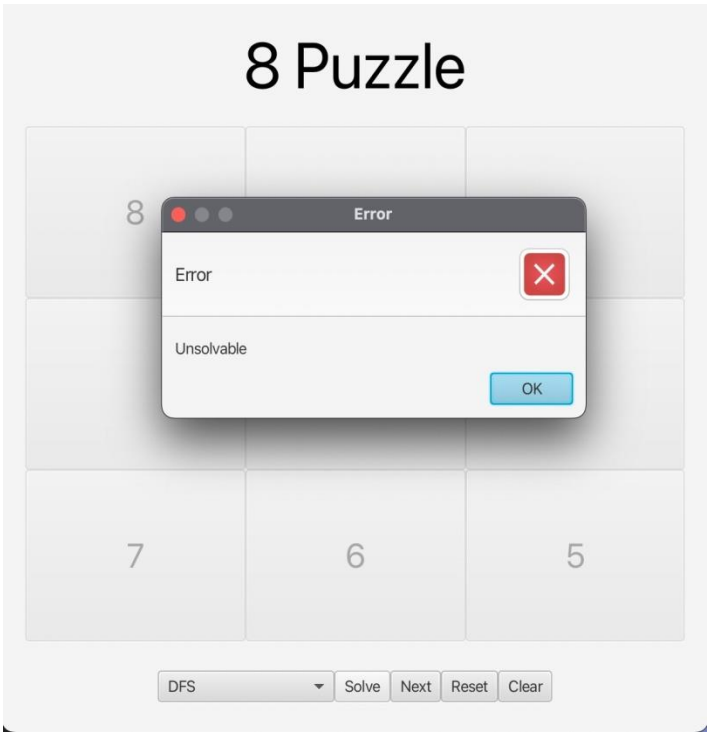

8 Puzzle
1
2
5
3
4
6
7
8
Select an algorithm
Solve
Next
Reset
Clear


8 Puzzle
1
2
3
4
5
6
7
8
BFS
Solve
Next
Reset
Clear


8 Puzzle
1
2
3
4
5
6
7
8
BFS
Solve
Next
Reset
Clear



Test case unsolvable:



Test Case:

```
1,2,5,3,4,0,6,7,8
           A* search(Manh):
           Success
           Path
125340678
120345678
102345678
012345678

           Cost
3
           Expanded
120345678, 102345678, 125340678, 012345678,
           Max depth
3
Execution time: 79 milliseconds
```

```
            A* search(Eucli)
            Success
            Path
125340678
120345678
102345678
012345678

            Cost
3
            Expanded
125340678, 102345678, 012345678, 120345678,
            Max depth
3
Execution time: 0 milliseconds
```

```
********************************************************************************
           BFS search
           Success
           Path
125340678
120345678
102345678
012345678

           Cost
3
           Expanded
125348607, 125304678, 012345678, 125374608, 125348670, 125034678, 125340678, 102345678, 120345678, 105324678,
           Max depth
3
Execution time: 0 milliseconds
```

```
**********************************************************************************
            DFS search
            Success
            Path
125340678
125348670
125348607
125348067
125048367
125408367
125480367
125487360
125487306
125487036
125087436
125807436
125870436
125876430
125876403
125876043
125076843
125706843
125760843
125763840
125763804
```

```
            Cost
157
            Expanded
312087546, 125634078, 312764085, 125630784, 125870436, 543218067, 254761803, 543087126, 431076825, 312764805, 125
            Max depth
157
Execution time: 16 milliseconds
**********************************************************************************
```

Test Case:

```
0,1,3,4,2,5,7,8,6
            A* search(Manh):
            Success
            Path
013425786
413025786
413725086
413725806
413725860
413720865
413702865
413762805
413762085
413062785
413602785
403612785
430612785
432610785
432615780
432615708
432615078
432015678
032415678
302415678
312405678
312045678
012345678
```

```
           Cost
22
           Expanded
213540786, 241536708, 410532786, 413268075, 412536708, 345082176, 123705846, 415802376, 412056738, 162438750, 413875026, 243185076, 512304786, 302415786,
           Max depth
22
Execution time: 184 milliseconds
```

```
A* search(Eucli)
Success
Path
013425786
413025786
413725086
413725806
413725860
413720865
413702865
413762805
413762085
413062785
413602785
403612785
430612785
432610785
432615780
432615708
432615078
432015678
032415678
302415678
312405678
312045678
012345678
```

```
Cost
22
Expanded
241503786, 345160728, 715032846, 412783605, 013462785, 120463758, 123856407, 243716058, 412530678, 182403765, 432815670, 102
Max depth
22
Execution time: 285 milliseconds
```

Test Cases:

```
1,2,0,3,4,5,6,7,8
            A* search(Manh):
            Success
            Path
120345678
102345678
012345678

            Cost
2
            Expanded
102345678, 012345678, 120345678,
            Max depth
2
Execution time: 64 milliseconds
********************************************************************************
            A* search(Eucli)
            Success
            Path
120345678
102345678
012345678

            Cost
2
            Expanded
102345678, 120345678, 012345678,
            Max depth
2
Execution time: 0 milliseconds
********************************************************************************
```

```
**************************************************************************
            BFS search
            Success
            Path
120345678
102345678
012345678

            Cost
2
            Expanded
102345678, 120345678, 012345678, 125340678,
            Max depth
2
Execution time: 8 milliseconds
**************************************************************************
```

```
            DFS search
            Success
            Path
120345678
125340678
125348670
125348607
125348067
125048367
125408367
125480367
125487360
125487306
125487036
125087436
125807436
125870436
125876430
125876403
125876043
125076843
125706843
125760843
125763840
125763804
125763084
125063784
125603784
```

```
           Cost
158
           Expanded
312706854, 254138607, 312045678, 250134678, 125763840, 543087126, 254613078, 543706812, 431605782, 312645078, 1
           Max depth
158
Execution time: 16 milliseconds
********************************************************************************

Process finished with exit code 0
```