**Class: STATE**

1. **State(r)**

3n^2 + 9n + 8  
O(n^2)

1. **setBoard(int r)**

3n^2 + 9n + 8  
O(n^2)

1. **genInitialState()**

19n + 5

O(n)

1. **getBlockCurrentRowCol()**

3n^2 + 3n+ 3

O(n^2)

1. **isGoalAchieved()**

2

O(1)

1. **calculateDistance()**
2. 15n^2 + 5n + 4
3. 15n^2 + 3n + 10
4. 15n^2 + 5n + 11

O(n^2)

1. **getBlockNumberCol()**

3n^2 + 3n + 2

O(n^2)

1. **getHeurisiticValue()**

9n^2 + 9n + 33 [maximum]

O(n^2)

1. **getRowPositionOfNumber()**

3n^2 + 3n + 3

O(n^2)

1. **howManyAboveGoalPosition()**
   1. 4n+22

O(n)

1. **howManyAboveBlockNumber()**
   1. 3n^2 + 7 n + 9

O(n^2)

1. **isBlockNumberTop()**

3n^2 + 4n + 15

O(n^2)

1. **isColumnEmptyOfGoal()**

2

O(1)

1. **isGoalSpaceEmpty()**

2

O(1)

1. **isTopZero()**

5

O(1)

1. **isTopNotZero()**

6

O(1)

1. **displayBoard()**

4n^2 + 8n + 3

O(n^2)

1. **getMoveThenMakeMove()**

3n^2 + 3n + 7

O(n^2)

1. **availablePositions()**

9n^2 + 9n + 1

O(n^2)

1. **numberUsedIn()**
   1. 3n+4

O(n)

1. **topValues()**
   1. 11n^2 + 7n + 1

O(n^2)

1. **topValuesDisjunctive()**
   1. 7n^2 + 3n + 1

O(n^2)

1. **operator=()**

3n^2 + 3n + 3

O(n^2)

1. **operator==()**

3n^2 + 3n + 2

O(n^2)

1. **destroyBoard()**

3n + 2

O(n)