CS571 AI LAB 01

14th August 2022

Ishita Singh 1901CS27
Kavya Goyal 1901CS30
Priyanka Sachan 1901CS43

Team Code: 1901cs30_1901cs27_1901cs43

OBJECTIVE

The objective of this assignment was to solve the 8-puzzle problem using the classical uninformed search algorithms namely BFS (Breadth First Search) and DFS (Depth First search).

ALGORITHM

The 8-puzzle problem involves a grid matrix with 1-8 numbers and a blank space where the blank B can be moved up, down, right or left to get the fixed desired target configuration. The classical BFS and DFS algorithms are used to generate the transition states from a randomly generated start state to find whether or not we can achieve the given target state

Initial State
//randomly generated
.
.
Final State

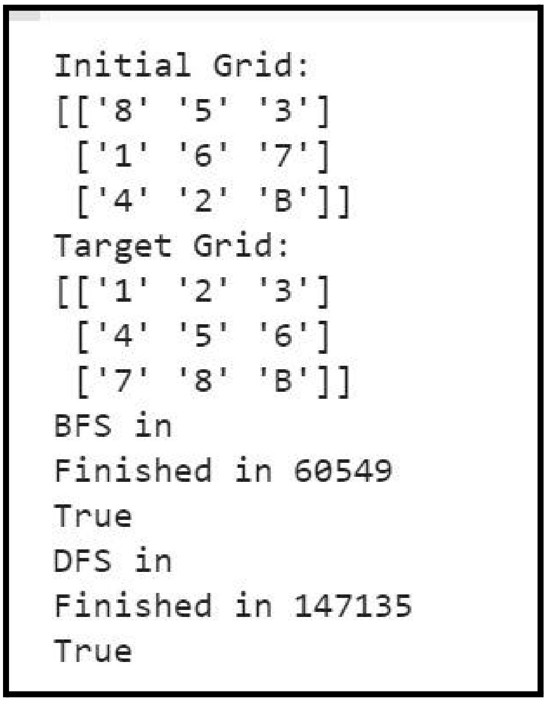
1	2	3
4	5	6
7	8	B

PERFORMING BFS and DFS

BFS or breadth first search means that all the children of a particular node are analysed first in a first in first out manner. For the purpose of this, a Python queue has been used. DFS or depth first search means that all the successive children of a particular node are recursively analysed in a last in a first out manner. For the purpose of this, a Python stack has been used. The queue/stack is initialised by the starting state and the following states are added based on whether they are visited or not.

SIMULATIONS AND TEST CASES

Case 1



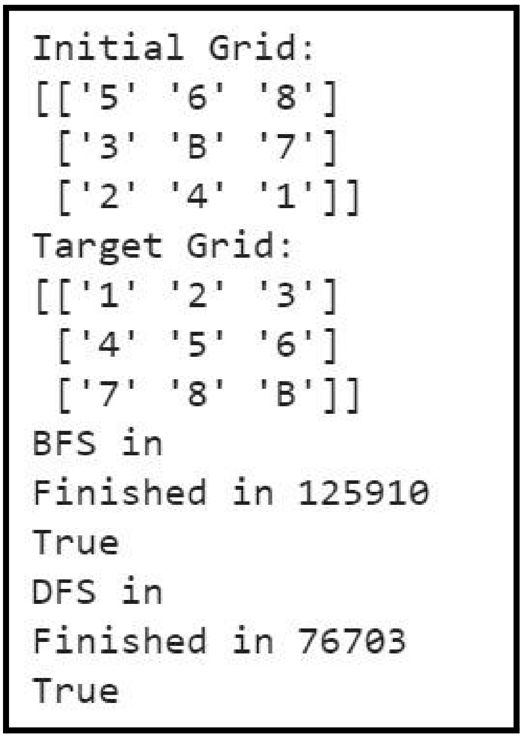
```
Initial Grid:
[['8' '5' '3']
 ['1' '6' '7']
 ['4' '2' 'B']]
Target Grid:
[['1' '2' '3']
 ['4' '5' '6']
 ['7' '8' 'B']]
BFS in
Finished in 60549
True
DFS in
Finished in 147135
True
```

The total possible states of the 88-puzzle -problem are 9! as at every space/position we have 9,8,7,...1 choices respectively.

The number of steps taken by BFS and DFS to reach the result are different depending on the depth of the target solution.

For cases where in the target state is much closer to the root of the graph, BFS performs much faster than DFS

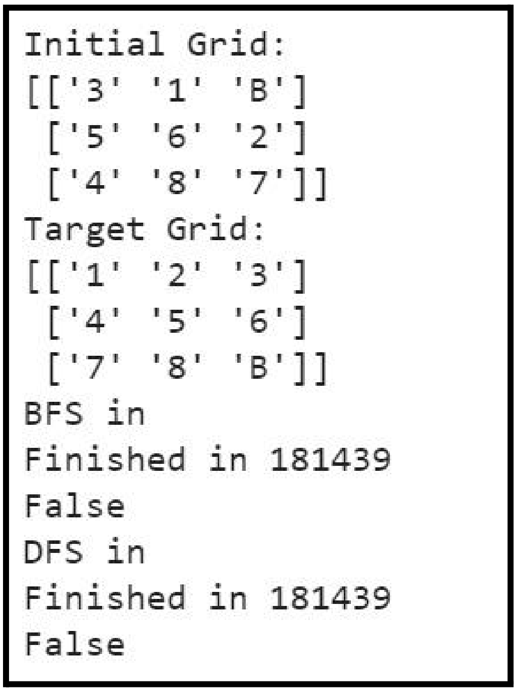
Case 2



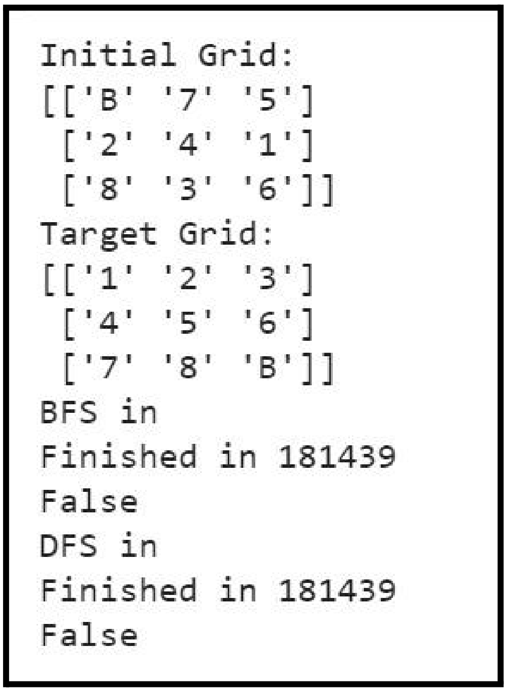
```
Initial Grid:
[['5' '6' '8']
 ['3' 'B' '7']
 ['2' '4' '1']]
Target Grid:
[['1' '2' '3']
 ['4' '5' '6']
 ['7' '8' 'B']]
BFS in
Finished in 125910
True
DFS in
Finished in 76703
True
```

For cases where in the target state is farther to the root of the graph, DFS performs faster than BFS

Case 3



```
Initial Grid:
[['3' '1' 'B']
 ['5' '6' '2']
 ['4' '8' '7']]
Target Grid:
[['1' '2' '3']
 ['4' '5' '6']
 ['7' '8' 'B']]
BFS in
Finished in 181439
False
DFS in
Finished in 181439
False
```



```
Initial Grid:
[['B' '7' '5']
 ['2' '4' '1']
 ['8' '3' '6']]
Target Grid:
[['1' '2' '3']
 ['4' '5' '6']
 ['7' '8' 'B']]
BFS in
Finished in 181439
False
DFS in
Finished in 181439
False
```

Out of these 9! = 362880, only half of the states are such that this final configuration is achieved. Therefore, there are 181440 states which do not return the final configuration, hence both the BFS and DFS run 181439 times and return a false as the result.