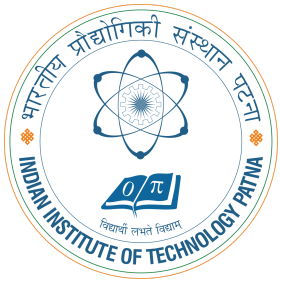
CS571: Artificial Intelligence Lab

Indian Institute of Technology Patna



ASSIGNMENT 1

DFS, BFS

Group Members

Kartik Kailas Mouli (2001CS35)
Rohit Ranjan (2001CS56)
Siddhant Kumar (2001CS70)

Q1. Compare Breadth First Search (BFS) and Depth First Search(DFS) with respect to the number of steps required to reach the solution if they are reachable.

When using BFS, we can always find the shortest path to the goal on a grid. But when using DFS, there's a chance we might find a longer path first, as long as we can actually reach the goal. If we can't reach the goal, both methods will go through a lot of states before stopping – specifically, about 9!/2 states.

However, if we think about the states we've already visited, BFS generally takes fewer steps to get to the goal. If the leaf node is closer to the target, then DFS performs better. Lower cost operations are handled better by BFS.

Q2. Comment on which algorithm will be faster and when, by mentioning proper intuition and examples.

BFS will work faster in the cases when the target node is close to the source node. That is, if the minimum number of steps, i.e, The cost to reach the target state is less, BFS will perform better.

Example:

Initial state:

1	2	4
B	6	8
7	3	5

Target state:

1	2	3
4	5	6
7	8	B

In this case, the minimum number of steps required is 17. Hence the number of iterations (states covered) taken by BFS will be 14809.

For the same initial and target state, DFS will cover more than 100000 states (this may vary depending on the order of the DFS traversal).

DFS will be faster when the target node is close to the leaf node. Starting off on the closest leaf node can also make all the difference.

Another example:

Initial state:

4	5	6
1	8	7
B	2	3

Target state:

1	2	3
4	5	6
7	8	B

BFS, on the other hand, needs to go through 134933 states to reach the target grid. So in some cases, DFS (steps 105786) does prove to be faster than BFS.

But in most scenarios, since choosing the right leaf node is random, BFS performs better than DFS.

Code Execution Examples:

```
Start grid is:
(4, 5, 6)
(1, 8, 7)
('B', 2, 3)

BFS:
No. of moves: 24
Stated Explored:  134933
Time taken by BFS: 0.70714

DFS:
No. of moves:  65128
Stated Explored:  105786
Time taken by DFS: 0.56145
PS C:\Users\karti> python -u "d:\College\Sem7\CS571 AI Lab\Assignment 1\1.py"

Start grid is:
(1, 2, 4)
('B', 6, 8)
(7, 3, 5)

BFS:
No. of moves: 17
Stated Explored:  14809
Time taken by BFS: 0.07851

DFS:
No. of moves:  56603
Stated Explored:  125054
Time taken by DFS: 0.72076
PS C:\Users\karti> python -u "d:\College\Sem7\CS571 AI Lab\Assignment 1\tempCodeRunnerFile.py"

Start grid is:
(4, 'B', 8)
(1, 5, 7)
(3, 2, 6)

Cannot reach the target.
```

```
Start grid is:
(1, 8, 'B')
(6, 3, 7)
(2, 5, 4)

Cannot reach the target.
PS C:\Users\karti> python -u "d:\College\Sem7\CS571 AI Lab\Assignment 1\1.py"

Start grid is:
(7, 'B', 6)
(5, 3, 2)
(1, 4, 8)

BFS:
No. of moves: 19
Stated Explored:  33519
Time taken by BFS: 0.18328

DFS:
No. of moves:  65973
Stated Explored:  100665
Time taken by DFS: 0.53326
PS C:\Users\karti> python -u "d:\College\Sem7\CS571 AI Lab\Assignment 1\1.py"

Start grid is:
(7, 4, 6)
(3, 5, 'B')
(8, 2, 1)

BFS:
No. of moves: 23
Stated Explored:  104681
Time taken by BFS: 0.56294

DFS:
No. of moves:  60983
Stated Explored:  75931
Time taken by DFS: 0.44522
```

Conclusion:

Overall, the results show that BFS generally produces fewer moves and explores fewer states compared to DFS, resulting in a more efficient solution in terms of both moves and exploration. However, the efficiency of each algorithm can vary depending on the specific initial state and the complexity of the puzzle.