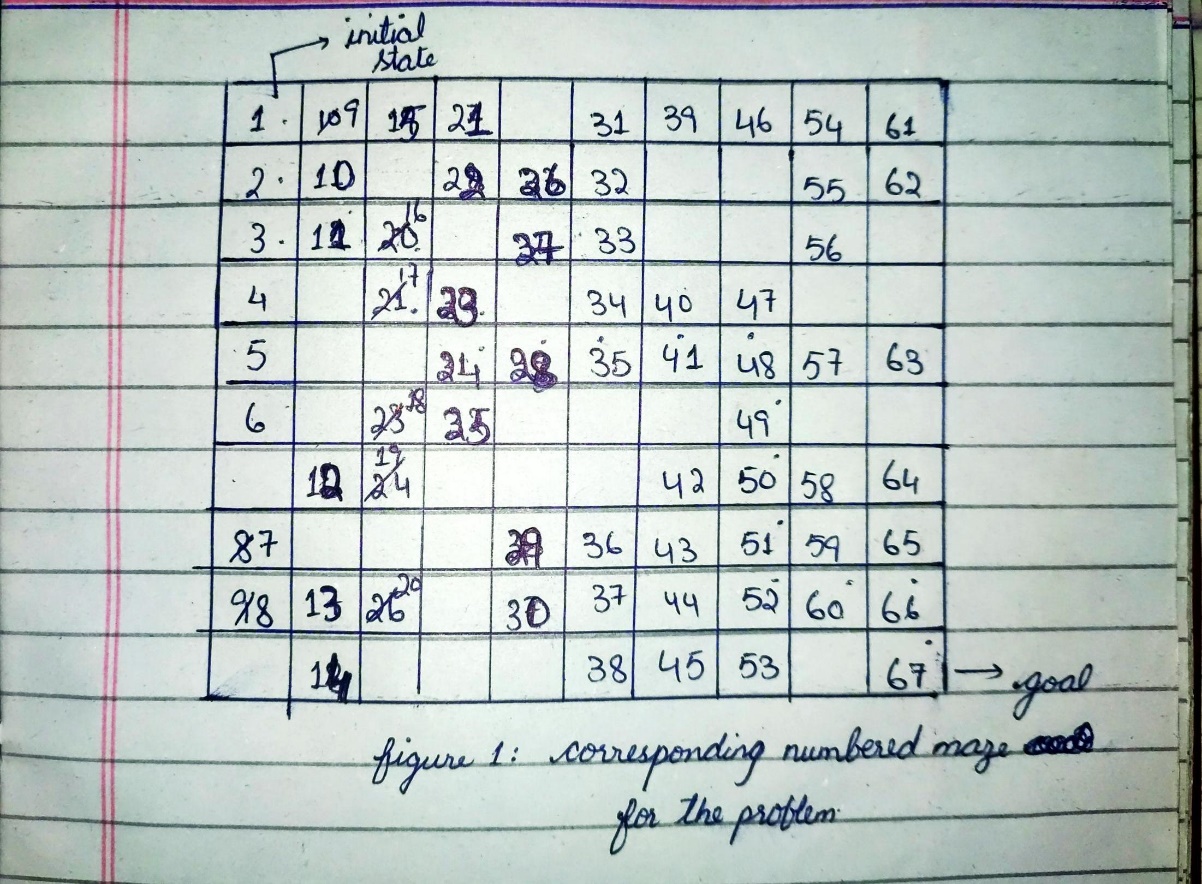
**Kulsoom khurshid**

**SP20-BCS-044**

**Assignment #1**

**Question 1)**

**Part a)**

To formulate this problem, following are the states and functions of the problem;

Initial state: The red box on the top left corner is the initial state.

Goal test: To verify if the robot is on the green box on the bottom right corner.

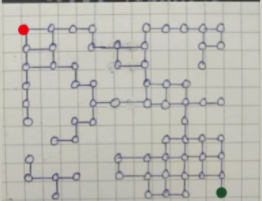
Successor function: To turn in north, south, east and west direction and keep moving in these directions.

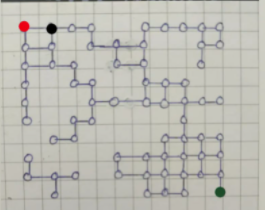
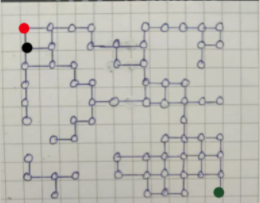
Cost function: The total cost will be from initial state to the current box or state. It is 1 unit per box.

**Part b)**

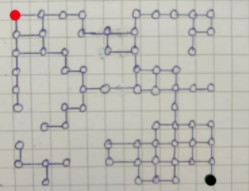
State space: There are total 67 boxes and 4 directions we can move. So, for the maze problems we compute state space by multiply total position with total directions. So, the state space will be 67 \* 4 = 268.

State space graph:



****

It keeps on expanding until it reaches destination.

**………………..**

**Part c)**

The robot is moving in four directions until it reaches its turning point. Following are the states and functions for this problem.

Initial state: The red box on the top left corner is the initial state.

Goal test: To verify if the robot is on the green box on the bottom right corner.

Successor function: To turn in north, south, east and west direction and keep moving in these directions.

Cost function: The total cost will be from initial state to the current box or state. It is 1 unit per box.

The only action that is taken is turning the robot in a direction. So we don’t need to keep its orientation because the robot will move in any one of the directories until it reaches it turning point.

**Part d)**

**BFS:**

Final path:

1 -> 2 -> 3 -> 11 -> 16 -> 17 -> 23 -> 24 -> 28 -> 35 -> 41 -> 48 -> 49 -> 50 -> 51 -> 52 -> 60 -> 66 -> 67

Traversed nodes:

1 -> 2 -> 9 -> 3 -> 10 -> 15 -> 4 -> 11 -> 21 -> 5 -> 16 -> 22 -> 6 -> 17 -> 26 -> 23 -> 27 -> 32 -> 24 -> 33 -> 31 -> 25 -> 28 -> 34 -> 39 -> 18 -> 35 -> 40 -> 46 -> 19 -> 41 -> 47 -> 54 -> 12 -> 48 -> 55 -> 61 -> 49 -> 57 -> 56 -> 62 -> 50 -> 63 -> 42 -> 51 -> 58 -> 43 -> 52 -> 59 -> 64 -> 36 -> 44 -> 53 -> 60 -> 65 -> 29 -> 37 -> 45 -> 66

**DFS:**

Final path:

1 -> 9 -> 15 -> 21 -> 22 -> 26 -> 32 -> 33 -> 34 -> 40 -> 47 -> 48 -> 49 -> 50 -> 58 -> 64 -> 65 -> 66 -> 67

Traversed nodes:

1 -> 9 -> 15 -> 21 -> 22-> 26 -> 32 -> 33 -> 34 -> 40 -> 47 -> 48 -> 57 -> 63 -> 49 -> 50 -> 58 -> 64 -> 65 -> 66

Analysis:

Both the algorithms final path has 19 boxes. But BFS traversed more boxes than DFS to reach the goal.

DFS is more suitable for searching when goal is far away from initial just like in this problem. DFS is better option in mazes and puzzles as BFS explore the neighbours as well where as DFS explore the path of a decision.