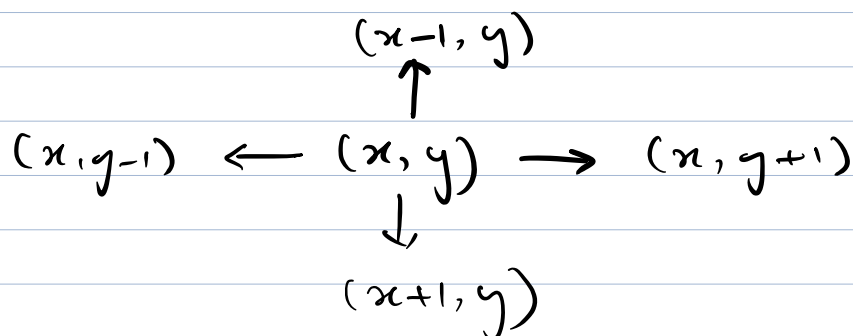
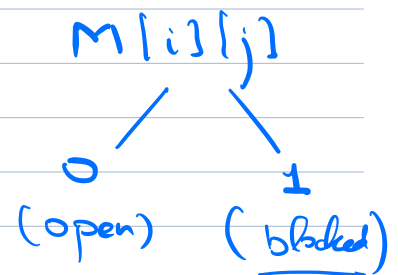
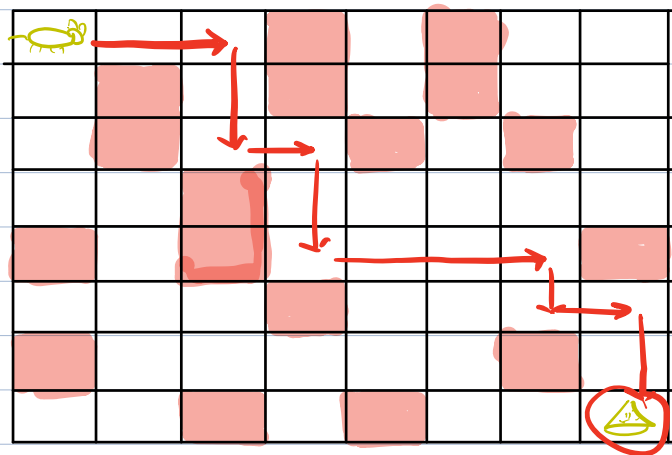


- 1) Rat in a maze
- 2) Extension
- 3) N Queen

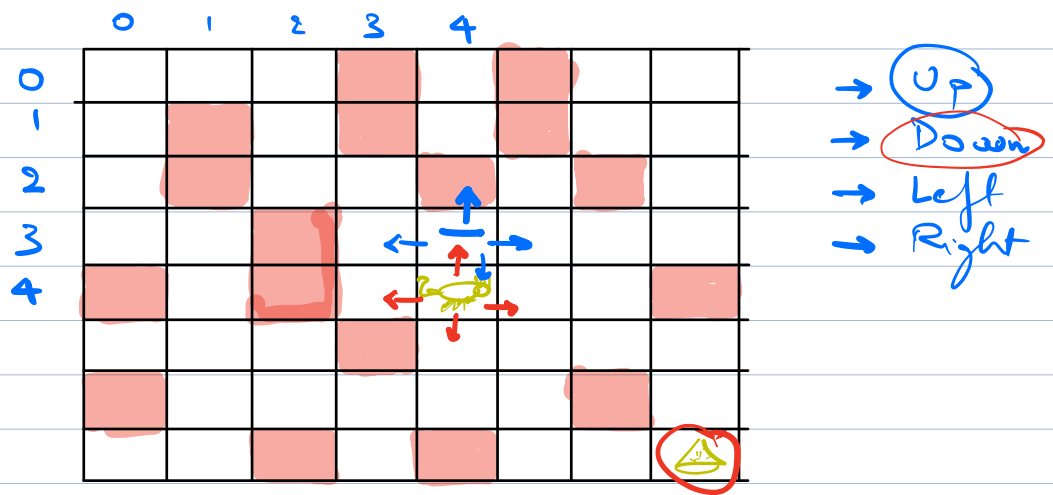
Given a maze (2D matrix) ($N \times M$)

- 1) Initial location of the mouse (x, y),
- 2) Location of cheese ($N-1, M-1$)

Return true if there exists a path from the initial location to the cheese



Solⁿ



(4,4)
↓
(3,4)
↓
(4,4)
↓
(3,4)
↓
(4,4)

Infinite

(Mask cells as visited)
 $M[i][j] = 2$

Code

```

boolean findPath (maze, N, M, x, y) {
    if ((x == N-1) && (y == M-1)) &
        return True;
    &
    if (x < 0 || x > N || y < 0 || y > M) &
        return false;
}
  
```

```

if ( maze[x][y] == 1 || maze[x][y] == 2 )

```

```

    return false;

```

```

    maze[x][y] = 2;

```

```

    bool result = findPath ( maze, N, M, x-1, y )

```

```

                        OR
    findPath ( maze, N, M, x+1, y )

```

```

                        OR
    findPath ( maze, N, M, x, y+1 )

```

```

                        OR
    findPath ( maze, N, M, x, y-1 );

```

```

    maze[x][y] = 0;

```

```

    return result;

```

```

}

```

T.C. ??

$(N \times M) ??$



Rat in a maze.

Given

start \Rightarrow S_i, S_j

End \Rightarrow E_i, E_j

Blocked \Rightarrow 1

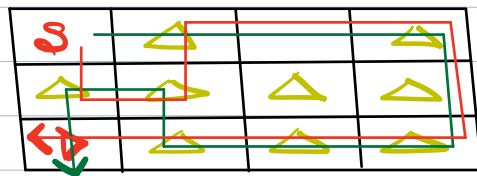
Empty \Rightarrow 0

Cheese \Rightarrow 2

\rightarrow (3) visited

Count the no. of paths from start

to end s.t. the rat can eat all the cheese present in the maze w/o stepping on the same cell twice. in one path.



$N \times M$

No. of cheese eaten in a path.

$==$

Total cheese in the maze.

9.9

$O(N \times M)$

Global \Rightarrow maze, N, M,

S_i, S_j, e_i, e_j , total cheese

curr cheese $= 0$

Code

int CountPath ($\overset{S_i}{\uparrow} x, \overset{S_j}{\uparrow} y$) {

if ($x < 0$ || $x > N$ || $y < 0$ || $y > M$) {

return 0;

}

if (maze[x][y] == 1 || maze[x][y] == 2) {

return 0;

}

if (x == ei && (y == ej)) {

if (currCheese == totalCheese) {
return 1;

}

else {

return 0;

}

}

if (maze[x][y] == 2) {
currCheese ++;

}

int temp = maze[x][y];
maze[x][y] = 3;

int ans = countPath(x+1, y)
+
countPath(x-1, y)
+
countPath(x, y+1)
+
countPath(x, y-1);

maze[x][y] = temp;

if (maze[x][y] == 2) {
currCheese --;

}

return ans;

}



N Queen Problem.

Google

FB

MS

Flipkart

Paypal

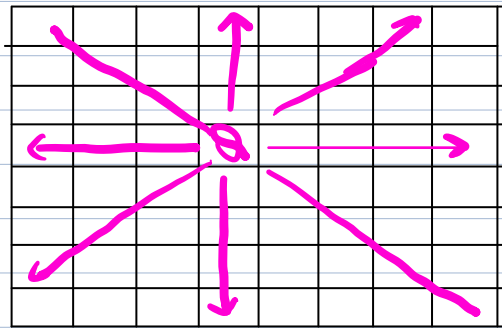
Allo

Given a $N \times N$ chessboard
N Queens.

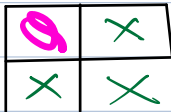
Arrange the queens in the board
st.

No queens targets any other
queen.

$N = 1$

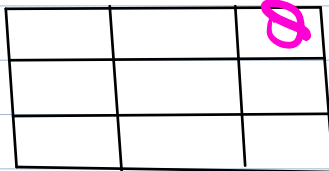


$N = 2$



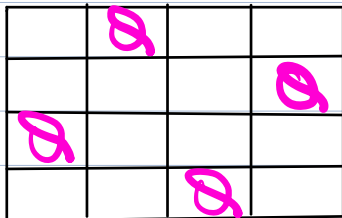
X

$N = 3$



X

$N = 4$



Code \Rightarrow (H.W.)

Thursday \Rightarrow Followup session of
Backtracking

1) Code of N Queen

\rightarrow 2) More problems (permutations)