135. Given an m x n grid and a ball at a starting cell, find the number of ways to move the ball out of the grid boundary in exactly N steps.

Program:

```
def findPaths(m, n, N, i, j):

    MOD = 10**9 + 7

    dp = [[0] * n for _ in range(m)]

    dp[i][j] = 1

    directions = [(0, 1), (1, 0), (0, -1), (-1, 0)]

    total_paths = 0

    for _ in range(N):

        temp = [[0] * n for _ in range(m)]

        for r in range(m):

            for c in range(n):

                for dr, dc in directions:

                    nr, nc = r + dr, c + dc

                    if 0 <= nr < m and 0 <= nc < n:

                        temp[nr][nc] = (temp[nr][nc] + dp[r][c]) % MOD

                    else:

                        total_paths = (total_paths + dp[r][c]) % MOD

        dp = temp

    return total_paths

print(findPaths(2, 2, 2, 0, 0))
```

Output:

TIME COMPLEXITY:-O(N *n*m)