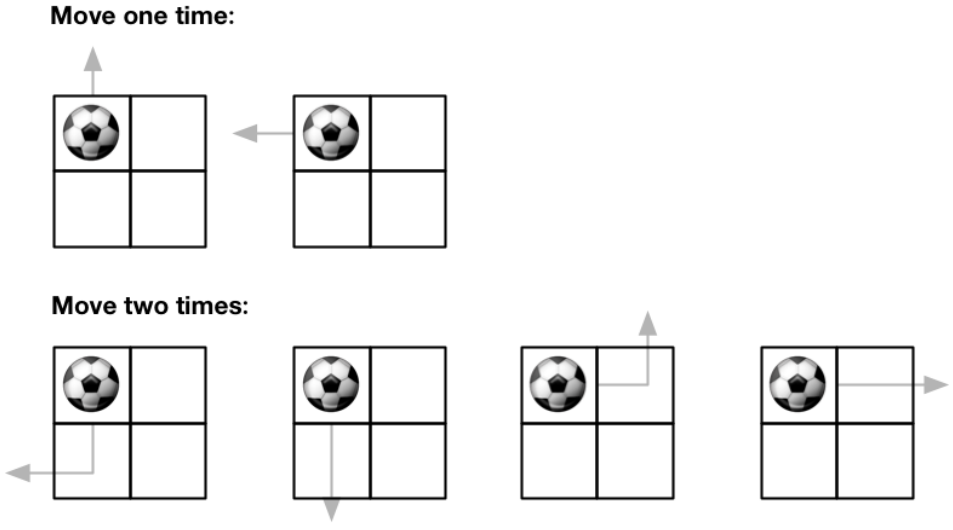
576. Out of Boundary Paths

Medium 2510 217 Add to List Share

There is an `m x n` grid with a ball. The ball is initially at the position `[startRow, startColumn]`. You are allowed to move the ball to one of the four adjacent cells in the grid (possibly out of the grid crossing the grid boundary). You can apply **at most** `maxMove` moves to the ball.

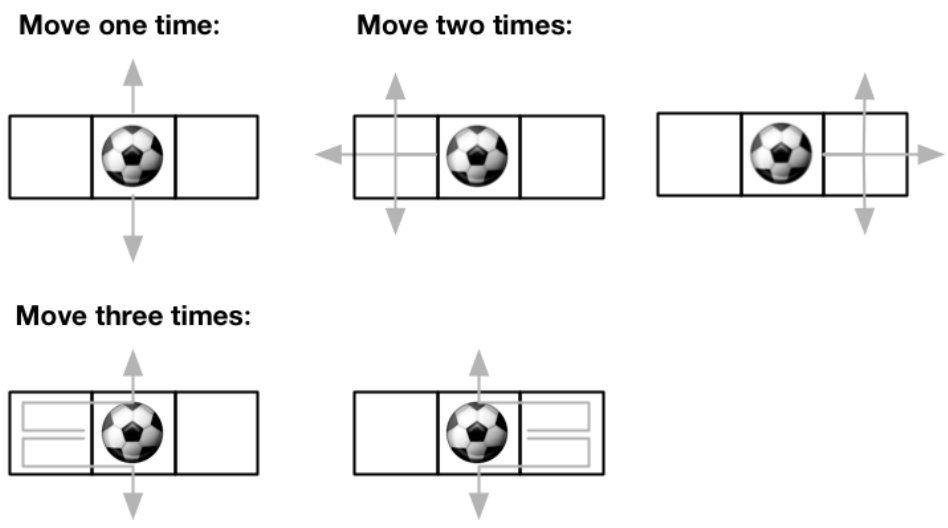
Given the five integers `m`, `n`, `maxMove`, `startRow`, `startColumn`, return the number of paths to move the ball out of the grid boundary. Since the answer can be very large, return it **modulo** $10^9 + 7$.

Example 1:




```
Input: m = 2, n = 2, maxMove = 2, startRow = 0, startColumn = 0
Output: 6
```

Example 2:




```
Input: m = 1, n = 3, maxMove = 3, startRow = 0, startColumn = 1
Output: 12
```

Constraints:

- `1 <= m, n <= 50`
- `0 <= maxMove <= 50`
- `0 <= startRow < m`
- `0 <= startColumn < n`

Accepted 97,284 Submissions 223,598

Seen this question in a real interview before? Yes No

Companies

C++ Autocomplete

```cpp
class Solution {
public:
    int memo[51][51][51];
    int mod = 1000000007;

    int findPathsH(int m, int n, int maxMove, int i, int j){
        if(i >= m || i < 0 || j >= n || j < 0){
            return 1;
        }

        if(maxMove <= 0){
            return 0;
        }

        if(memo[i][j][maxMove] != -1){
            return memo[i][j][maxMove];
        }

        long res = 0;
        res += findPathsH(m, n, maxMove-1, i-1, j);
        res += findPathsH(m, n, maxMove-1, i+1, j);
        res += findPathsH(m, n, maxMove-1, i, j-1);
        res += findPathsH(m, n, maxMove-1, i, j+1);

        return memo[i][j][maxMove] = res % mod;
    }

    int findPaths(int m, int n, int maxMove, int startRow, int startColumn) {
        memset(memo, -1, sizeof(memo));
        return findPathsH(m, n, maxMove, startRow, startColumn);
    }
};
```

Your previous code was restored from your local storage. Reset t

Testcase Run Code Result Debugger

Accepted Runtime: 5 ms

Your input
2
2

Output 6 Diff

Expected 6