Description | Editorial | Solutions (760) | Submissions

## 2328. Number of Increasing Paths in a Grid

Hint

Hard | 1.5K | 33

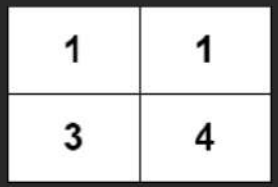
Companies

You are given an `m x n` integer matrix `grid`, where you can move from a cell to any adjacent cell in all `4` directions.

Return *the number of **strictly increasing** paths in the grid such that you can start from **any** cell and end at **any** cell.* Since the answer may be very large, return it **modulo** `10^9 + 7`.

Two paths are considered different if they do not have exactly the same sequence of visited cells.

**Example 1:**




```
Input: grid = [[1,1],[3,4]]
Output: 8
Explanation: The strictly increasing paths are:
- Paths with length 1: [1], [1], [3], [4].
- Paths with length 2: [1 -> 3], [1 -> 4], [3 -> 4].
- Paths with length 3: [1 -> 3 -> 4].
The total number of paths is 4 + 3 + 1 = 8.
```

**Example 2:**

```
Input: grid = [[1],[2]]
Output: 3
Explanation: The strictly increasing paths are:
- Paths with length 1: [1], [2]
```

Java | Auto

```java
import java.util.Arrays;

class Solution {
    int mod = (int) Math.pow(10, 9) + 7;

    public int countPaths(int[][] grid) {
        int rows = grid.length;
        int cols = grid[0].length;

        int[][] dp = new int[rows][cols];
        for (int[] row : dp) {
            Arrays.fill(row, -1);
        }

        for (int i = 0; i < rows; i++) {
            for (int j = 0; j < cols; j++) {
                if (dp[i][j] == -1) {
```

Testcase | Result

**Accepted** Runtime: 0 ms

Case 1 | Case 2

Input

grid =
`[[1,1],[3,4]]`

Output

8

Expected

8

Console | Run | Submit

Description | Editorial | Solutions (10.2K) | Submissions

## 217. Contains Duplicate

Easy | 9.6K | 1.2K

Companies

Given an integer array `nums`, return `true` if any value appears **at least twice** in the array, and return `false` if every element is distinct.

**Example 1:**

```
Input: nums = [1,2,3,1]
Output: true
```

**Example 2:**

```
Input: nums = [1,2,3,4]
Output: false
```

**Example 3:**

```
Input: nums = [1,1,1,3,3,4,3,2,4,2]
Output: true
```

**Constraints:**

- `1 <= nums.length <= 10^5`
- `-10^9 <= nums[i] <= 10^9`

Accepted 2.9M | Submissions 4.7M | Acceptance Rate 61.3%

Java | Auto

```java
class Solution {
    public boolean containsDuplicate(int[] nums) {
        HashSet<Integer> set = new HashSet<>();
        for (int i = 0; i < nums.length; i++) {
                if (set.contains(nums[i])) {
                    return true;
                }
                set.add(nums[i]);
        }
        return false;
    }
}
```

Testcase | Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

nums =
[1,2,3,1]

Output

true

Expected

true

Console | Run | Submit