Unique Paths in a Grid

You are given a matrix **grid** of **n** x **m** size consisting of values **0** and **1**. A value of **1** means that you can enter that cell and **0** implies that entry to that cell is not allowed.

You start at the upper-left corner of the grid **(1, 1)** and you have to reach the bottom-right corner **(n, m)** such that you can only move in the right or down direction from every cell.

Your task is to calculate the total number of ways of reaching the target **modulo (10^9+7)**.
Note: The **first (1, 1)** and **last cell (n, m)** of the grid can also be **0**

Example 1:

```
Input:
n = 3, m = 3
grid[][] = {{1, 1, 1};
            {1, 0, 1};
            {1, 1, 1}}
Output:
2
Explanation:
1 1 1
1 0 1
1 1 1
This is one possible path.
1 1 1
1 0 1
1 1 1
This is another possible path.
```

Example 2:

```
Input:
n = 1, m = 3
grid = {{1, 0, 1}}
Output :
0
Explanation:
There is no possible path to reach the end.
```

Your Task:
You don't need to read input or print anything. Your task is to complete the function **uniquePaths()** which takes 2 integers n, and m, and a matrix of size n*m as input and returns the number of unique paths from cell (1,1) to (n,m) modulo (10^9+7)

Expected Time Complexity: O(n*m)
Expected Auxiliary Space: O(n*m)

Constraints:
$1 \leq n,m \leq 10^3$