Problem 2132: Stamping the Grid

Problem Information

Difficulty: Hard
Acceptance Rate: 34.52%
Paid Only: No
Tags: Array, Greedy, Matrix, Prefix Sum

Problem Description

You are given an `m x n` binary matrix `grid` where each cell is either `0` (empty) or `1` (occupied).

You are then given stamps of size `stampHeight x stampWidth`. We want to fit the stamps such that they follow the given **restrictions** and **requirements** :

1. Cover all the **empty** cells. 2. Do not cover any of the **occupied** cells. 3. We can put as **many** stamps as we want. 4. Stamps can **overlap** with each other. 5. Stamps are not allowed to be **rotated**. 6. Stamps must stay completely **inside** the grid.

Return `true` _if it is possible to fit the stamps while following the given restrictions and requirements. Otherwise, return_ `false`.

Example 1:

![](https://assets.leetcode.com/uploads/2021/11/03/ex1.png)

Input: grid = [[1,0,0,0],[1,0,0,0],[1,0,0,0],[1,0,0,0],[1,0,0,0]], stampHeight = 4, stampWidth = 3 **Output:** true **Explanation:** We have two overlapping stamps (labeled 1 and 2 in the image) that are able to cover all the empty cells.

Example 2:

![](https://assets.leetcode.com/uploads/2021/11/03/ex2.png)

Input: grid = [[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]], stampHeight = 2, stampWidth = 2 **Output:** false **Explanation:** There is no way to fit the stamps onto all the empty cells without the stamps going outside the grid.

Constraints:

* `m == grid.length` * `n == grid[r].length` * `1 <= m, n <= 105` * `1 <= m * n <= 2 * 105` * `grid[r][c]` is either `0` or `1`. * `1 <= stampHeight, stampWidth <= 105`

Code Snippets

C++:

```
class Solution {
public:
bool possibleToStamp(vector<vector<int>>& grid, int stampHeight, int
stampWidth) {

}
};
```

Java:

```
class Solution {
public boolean possibleToStamp(int[][] grid, int stampHeight, int stampWidth)
{

}
}
```

Python3:

```
class Solution:
def possibleToStamp(self, grid: List[List[int]], stampHeight: int,
stampWidth: int) -> bool:
```