Problem 1591: Strange Printer II

Problem Information

Difficulty: Hard
Acceptance Rate: 60.48%
Paid Only: No
Tags: Array, Graph, Topological Sort, Matrix

Problem Description

There is a strange printer with the following two special requirements:

* On each turn, the printer will print a solid rectangular pattern of a single color on the grid. This will cover up the existing colors in the rectangle. * Once the printer has used a color for the above operation, **the same color cannot be used again**.

You are given a `m x n` matrix `targetGrid`, where `targetGrid[row][col]` is the color in the position `(row, col)` of the grid.

Return `true` _if it is possible to print the matrix_`targetGrid` _,__otherwise, return_`false`.

Example 1:

Input: targetGrid = [[1,1,1,1],[1,2,2,1],[1,2,2,1],[1,1,1,1]] **Output:** true

Example 2:

Input: targetGrid = [[1,1,1,1],[1,1,3,3],[1,1,3,4],[5,5,1,4]] **Output:** true

Example 3:

Input: targetGrid = [[1,2,1],[2,1,2],[1,2,1]] **Output:** false **Explanation:** It is impossible to form targetGrid because it is not allowed to print the same color in different turns.

Constraints:

* `m == targetGrid.length` * `n == targetGrid[i].length` * `1 <= m, n <= 60` * `1 <= targetGrid[row][col] <= 60`

Code Snippets

C++:

```
class Solution {
public:
bool isPrintable(vector<vector<int>>& targetGrid) {

}
};
```

Java:

```
class Solution {
public boolean isPrintable(int[][] targetGrid) {

}
}
```

Python3:

```
class Solution:
def isPrintable(self, targetGrid: List[List[int]]) -> bool:
```