Problem 1034: Coloring A Border

Problem Information

Difficulty: Medium
Acceptance Rate: 50.60%
Paid Only: No
Tags: Array, Depth-First Search, Breadth-First Search, Matrix

Problem Description

You are given an `m x n` integer matrix `grid`, and three integers `row`, `col`, and `color`. Each value in the grid represents the color of the grid square at that location.

Two squares are called **adjacent** if they are next to each other in any of the 4 directions.

Two squares belong to the same **connected component** if they have the same color and they are adjacent.

The **border of a connected component** is all the squares in the connected component that are either adjacent to (at least) a square not in the component, or on the boundary of the grid (the first or last row or column).

You should color the **border** of the **connected component** that contains the square `grid[row][col]` with `color`.

Return _the final grid_.

Example 1:

Input: grid = [[1,1],[1,2]], row = 0, col = 0, color = 3 **Output:** [[3,3],[3,2]]

Example 2:

Input: grid = [[1,2,2],[2,3,2]], row = 0, col = 1, color = 3 **Output:** [[1,3,3],[2,3,3]]

Example 3:

Input: grid = [[1,1,1],[1,1,1],[1,1,1]], row = 1, col = 1, color = 2 **Output:** [[2,2,2],[2,1,2],[2,2,2]]

Constraints:

* `m == grid.length` * `n == grid[i].length` * `1 <= m, n <= 50` * `1 <= grid[i][j], color <= 1000` * `0 <= row < m` * `0 <= col < n`

Code Snippets

C++:

```
class Solution {
public:
vector<vector<int>> colorBorder(vector<vector<int>>& grid, int row, int col,
int color) {

}
};
```

Java:

```
class Solution {
public int[][] colorBorder(int[][] grid, int row, int col, int color) {

}
}
```

Python3:

```
class Solution:
def colorBorder(self, grid: List[List[int]], row: int, col: int, color: int)
-> List[List[int]]:
```