Problem Title: Flood Fill Algorithm

Company Tag: Asked by **Facebook**

Problem Scenario:

You're given an image represented as a 2D matrix of characters, where each character represents a pixel color. You're also given the coordinates of a pixel (row, col) and a new color `C`.

Your task is to perform a **flood fill** operation starting from the given pixel: change the color of the pixel and **all connected pixels** that have the **same original color** to the new color `C`.

Pixels are connected 4-directionally (up, down, left, right — no diagonals).

Input Format:

- A 2D character matrix `image[n][m]` representing the image
- Two integers `sr` and `sc` representing the row and column of the starting pixel
- A character `C` representing the new color

Output Format:

- The updated image matrix after applying the flood fill algorithm.

Example:

Input:

```
image = [
  ['B', 'B', 'W'],
  ['W', 'W', 'W'],
  ['W', 'W', 'W'],
  ['B', 'B', 'B']
]
sr = 2, sc = 2
C = 'G'
```

Output:

```
[
  ['B', 'B', 'G'],
  ['G', 'G', 'G'],
```

```
  ['G', 'G', 'G'],
  ['B', 'B', 'B']
]
```

Explanation:

The pixel at (2, 2) is originally `'W'`. The flood fill algorithm changes all connected `'W'` pixels (including itself) to `'G'`. Only the `'W'` pixels that are connected in four directions are changed. `'B'` pixels are left unchanged.

Constraints:

- `1 <= rows, cols <= 100`
- `C` is an uppercase character.
- The original image contains only uppercase characters.

Suggested Approaches:

- DFS (Depth First Search)
- BFS (Breadth First Search)

Practice Links:

- 🔗 LeetCode – Flood Fill
- 🔗 GeeksforGeeks – Flood Fill Algorithm

Video Explanation (Recommended):

- 📺 YouTube - Flood Fill Algorithm (DFS & BFS)