733. Flood Fill

Solved

Easy | Topics | Companies | Hint

You are given an image represented by an `m x n` grid of integers `image`, where `image[i][j]` represents the pixel value of the image. You are also given three integers `sr`, `sc`, and `color`. Your task is to perform a **flood fill** on the image starting from the pixel `image[sr][sc]`.

To perform a **flood fill**:

1. Begin with the starting pixel and change its color to `color`.
2. Perform the same process for each pixel that is **directly adjacent** (pixels that share a side with the original pixel, either horizontally or vertically) and shares the **same color** as the starting pixel.
3. Keep **repeating** this process by checking neighboring pixels of the *updated* pixels and modifying their color if it matches the original color of the starting pixel.
4. The process **stops** when there are **no more** adjacent pixels of the original color to update.

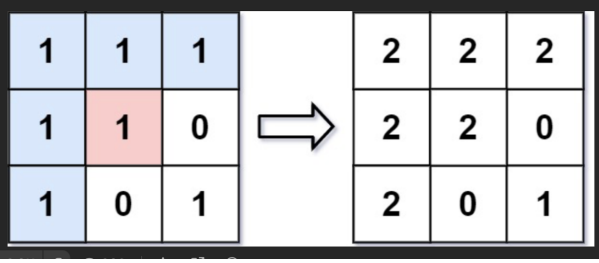
Return the **modified** image after performing the flood fill.

Example 1:

Input: `image = [[1,1,1],[1,1,0],[1,0,1]], sr = 1, sc = 1, color = 2`

Output: `[[2,2,2],[2,2,0],[2,0,1]]`

Explanation:



Python3 • Auto

```python
class Solution:
    def floodFill(self, image: List[List[int]], sr: int, sc: int, color: int) -> List[List[int]]:
        m = len(image) # row
        n = len(image[0]) # col
        origin_color = image[sr][sc]
        if color == origin_color:
            return image

        def dfs(r, c):
            if image[r][c] == origin_color:
                image[r][c] = color

                if r -1 >= 0:
                    dfs(r-1, c)
                if r + 1 < m:
                    dfs(r+1, c)
                if c -1 >= 0:
                    dfs(r, c-1)
                if c + 1 < n:
                    dfs(r, c+1)

        dfs(sr, sc)
        return image
```

Ln 20, Col 30 Saved

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

image =

`[[1,1,1],[1,1,0],[1,0,1]]`

#amazon
#DFS