

Majority Element

Cousins in Binary Tree

Week 2: May 8th–May 14th
Problems appear at midnight, Pacific Time

Check If It Is a Straight Line

Valid Perfect Square

Find the Town Judge

Flood Fill

Single Element in a Sorted ...

Week 3: May 15th–May 21st
The first problem for this section is ...

Week 4: May 22nd–May 28th
The first problem for this section is ...

Previous

Next

Go to Discuss

Flood Fill

Solution

An `image` is represented by a 2-D array of integers, each integer representing the pixel value of the image (from 0 to 65535).

Given a coordinate `(sr, sc)` representing the starting pixel (row and column) of the flood fill, and a pixel value `newColor`, "flood fill" the image.

To perform a "flood fill", consider the starting pixel, plus any pixels connected 4-directionally to the starting pixel of the same color as the starting pixel, plus any pixels connected 4-directionally to those pixels (also with the same color as the starting pixel), and so on. Replace the color of all of the aforementioned pixels with the `newColor`.

At the end, return the modified image.

Example 1:

Input:

```
image = [[1,1,1],[1,1,0],[1,0,1]]
sr = 1, sc = 1, newColor = 2
```

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

Explanation:
The starting pixel and its four adjacent pixels are colored with the new color.
Note the bottom corner is not colored 2, because it is not 4-directionally connected to the starting pixel.

Note:

- The length of `image` and `image[0]` will be in the range `[1, 50]`.
- The given starting pixel will satisfy `0 <= sr < image.length` and `0 <= sc < image[0].length`.
- The value of each color in `image[i][j]` and `newColor` will be an integer in `[0, 65535]`.

Show Hint #1

Java

```
1 class Solution {
2     public int[][] floodFill(int[][] image, int sr, int sc, int newColor) {
3         if(image.length==0 || image[0].length==0) return image;
4         public int[][] floodFill(int[][] image, int sr, int sc, int newColor) {
5             if(image.length==0 || image[sr][sc]==newColor) return image;
6             dfs(image,sr,sc,newColor,image[sr][sc]);
7             return image;
8         }
9         public void dfs(int[][] image,int r,int c,int newColor,int firstColor){
10             if(r<0 || r>=image.length || c<0 || c>=image[r].length){
11                 return;
12             }
13             if(image[r][c]==firstColor){
14                 image[r][c]=newColor;
15                 dfs(image,r-1,c,newColor,firstColor);
16                 dfs(image,r,c-1,newColor,firstColor);
17                 dfs(image,r+1,c,newColor,firstColor);
18                 dfs(image,r,c+1,newColor,firstColor);
19             }
20         }
21     }
22 }
```

Custom Testcase (Contribute)

Run Code

Submit