Flood fill Algorithm

An `image` is represented by a 2-D array of integers, each integer representing the pixel value of the image.

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.

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: From the center of the image
(with position (sr, sc) = (1, 1)), all
pixels connected by a path of the same color
as the starting pixel 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.
```

Your Task:
You don't need to read or print anyhting. Your task is to complete the function **floodFill()** which takes image, sr, sc and newColor as input paramater and returns the image after flood filling.

Expected Time Compelxity: O(n*m)
Expected Space Complexity: O(n*m)

Constraints:
1 <= n <= m <= 100
0 <= pixel values <= 10
0 <= sr <= (n-1)
0 <= sc <= (m-1)