```cpp
class Solution {
public:
    void DFS(vector<vector<int>>& image, int sr, int sc, int color,
             int startColor) {
        // Process stops when there are no more adjacent pixels of the original
        // color to update.
        if (sr < 0 || sr >= image.size() || sc < 0 || sc >= image[0].size() ||
            image[sr][sc] != startColor) {
            return;
        }
        // Change the color of the current pixel
        image[sr][sc] = color;

        // 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. Vertically
        DFS(image, sr - 1, sc, color, startColor);
        DFS(image, sr + 1, sc, color, startColor);
        // Horizontally
        DFS(image, sr, sc - 1, color, startColor);
        DFS(image, sr, sc + 1, color, startColor);
    }

    vector<vector<int>> floodFill(vector<vector<int>>& image, int sr, int sc,
                                  int color) {
        int startColor = image[sr][sc];
        // Call to DFS
        if (startColor != color) {
            DFS(image, sr, sc, color, startColor);
        }
        // Return the modified image after performing the flood fill
        return image;
    }
};
```