

## 832 翻转图像

Label: 二维数组

给定一个二进制矩阵 A，我们先水平翻转图像，然后反转图像并返回结果。

水平翻转图片就是将图片的每一行都进行翻转，即逆序。例如，水平翻转  $[1, 1, 0]$  的结果是  $[0, 1, 1]$ 。

反转图片的意思是图片中的 0 全部被 1 替换，1 全部被 0 替换。例如，反转  $[0, 1, 1]$  的结果是  $[1, 0, 0]$ 。

输入:  $[[1,1,0],[1,0,1],[0,0,0]]$

输出:  $[[1,0,0],[0,1,0],[1,1,1]]$

解释: 首先翻转每一行:  $[[0,1,1],[1,0,1],[0,0,0]]$ ;

然后反转图片:  $[[1,0,0],[0,1,0],[1,1,1]]$

输入:  $[[1,1,0,0],[1,0,0,1],[0,1,1,1],[1,0,1,0]]$

输出:  $[[1,1,0,0],[0,1,1,0],[0,0,0,1],[1,0,1,0]]$

解释: 首先翻转每一行:  $[[0,0,1,1],[1,0,0,1],[1,1,1,0],[0,1,0,1]]$ ;

然后反转图片:  $[[1,1,0,0],[0,1,1,0],[0,0,0,1],[1,0,1,0]]$

- 遍历 + 栈

```
class Solution {
    public int[][] flipAndInvertImage(int[][] image) {
        for (int i = 0; i < image.length; i++) {
            Stack<Integer> stack = new Stack<>();
            for (int j = 0; j < image[0].length; j++) { //push
                stack.push(image[i][j]);
            }
            for (int j = 0; j < image[0].length; j++) { //pop
                image[i][j] = 1 - stack.pop();
            }
        }
        return image;
    }
}
```

- 类似 空间换时间

```
class Solution {
    public int[][] flipAndInvertImage(int[][] A) {
        int x=A.length,y=A[0].length;
        int[][] B = new int[x][y];
        for (int i=0;i<x;i++){
            for (int j=0;j<y;j++){
                B[i][y-j-1]=1-A[i][j];    //从后往前赋值，同时取反
            }
        }
        return B;
    }
}
```

- 时间换空间 + 双指针

```
class Solution {
    public int[][] flipAndInvertImage(int[][] image) {
        int n = image.length;
        for (int i = 0; i < n; i++) {
            int left = 0, right = n - 1;
            while (left < right) {
                if (image[i][left] == image[i][right]) {
                    image[i][left] ^= 1;
                    image[i][right] ^= 1;
                }
                left++;
                right--;
            }

            if (left == right) { // 每行个数有奇数
                image[i][left] ^= 1;
            }
        }
        return image;
    }
}
```