

```

1  /*
2  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。
3  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,
4  1, 1]。
5  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
6  是 [1, 0, 0]。
7  示例 1:
8  输入: [[1,1,0],[1,0,1],[0,0,0]]
9  输出: [[1,0,0],[0,1,0],[1,1,1]]
10 解释: 首先翻转每一行: [[0,1,1],[1,0,1],[0,0,0]];
11      然后反转图片: [[1,0,0],[0,1,0],[1,1,1]]
12 示例 2:
13 输入: [[1,1,0,0],[1,0,0,1],[0,1,1,1],[1,0,1,0]]
14 输出: [[1,1,0,0],[0,1,1,0],[0,0,0,1],[1,0,1,0]]
15 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]];
16      然后反转图片: [[1,1,0,0],[0,1,1,0],[0,0,0,1],[1,0,1,0]]
17 说明:
18     1 <= A.length = A[0].length <= 20
19     0 <= A[i][j] <= 1
20 来源: 力扣 (LeetCode)
21 链接: https://leetcode-cn.com/problems/flipping-an-image
22 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
23 */

```

#### • 方法一:暴力法

```

1  class Solution
2  {
3      public:
4          vector<vector<int>> flipAndInvertImage(vector<vector<int>>& A)
5          {
6              vector<vector<int>> ret_val;
7              vector<int> temp;
8              int i = 0;
9              int j = 0;
10             for(i=0;i<A.size();i++)
11             {
12                 temp = A[i];
13                 reverse(temp.begin(), temp.end());
14                 for(j=0;j<temp.size();j++)
15                 {
16                     temp[j] = !temp[j];
17                 }
18                 ret_val.push_back(temp);
19             }
20             return ret_val;
21         }
22     };
23
24     /*
25     执行结果:

```

```
26 | 通过
27 | 显示详情
28 | 执行用时 :32 ms, 在所有 C++ 提交中击败了17.27% 的用户
29 | 内存消耗 :9.5 MB, 在所有 C++ 提交中击败了68.44%的用户
30 | */
```

---

AlimyBreak  
2019.09.03