

73-矩阵置零

题述

力扣 学习 题库 讨论 竞赛 求职 商店

题目描述

评论 (663)

题解 (924)

提交记录

73. 矩阵置零

难度 中等

547

收藏

分享

切换为英文

接收动态

反馈

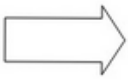
给定一个 $m \times n$ 的矩阵，如果一个元素为 0，则将其所在行和列的所有元素都设为 0。请使用原地算法。

进阶：

- 一个直观的解决方案是使用 $O(mn)$ 的额外空间，但这并不是一个好的解决方案。
- 一个简单的改进方案是使用 $O(m + n)$ 的额外空间，但这仍然不是最好的解决方案。
- 你能想出一个仅使用常量空间的解决方案吗？

示例 1:

1	1	1
1	0	1
1	1	1



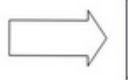
1	0	1
0	0	0
1	0	1

输入: matrix = [[1,1,1],[1,0,1],[1,1,1]]

输出: [[1,0,1],[0,0,0],[1,0,1]]

示例 2:

0	1	2	0
3	4	5	2
1	3	1	5



0	0	0	0
0	4	5	0
0	3	1	0

输入: matrix = [[0,1,2,0],[3,4,5,2],[1,3,1,5]]

输出: [[0,0,0,0],[0,4,5,0],[0,3,1,0]]

浅析

在计算机科学中，一个原地算法 (in-place algorithm) 是一种使用小的，固定数量的额外之空间来转换资料的算法。当算法执行时，输入的资料通常会被要输出的部分覆盖掉。不是原地算法有时候称为非原地 (not-in-place) 或不得其所 (out-of-place)。

可以先画个图分析一下

利用两个flag记录第一行第一列是否有0

遍历除第一行第一列以外的矩阵元素，如果有元素为0，则将对应的第一行第一列的位置置为0

同样遍历除第一行第一列以外的矩阵元素，如果matrix[i][j]所在位置的第一行第一列有任意一个位置为0

那么哪个位置将被置为0

最后再根据两个flag决定是否将第一行第一列置为0

代码

```
class Solution {
public:
    void setZeroes(vector<vector<int>>& matrix)
    {
        if(matrix.size()==0 || matrix[0].size()==0) return;
        bool rowFlag=false,colFlag=false;    //两个flag记录第一行第一列是否有0
        int rows=matrix.size();
        int cols=matrix[0].size();

        //看第一列是否有0
        for(int i=0;i<rows;++i)
        {
            if(matrix[i][0]==0)
            {
                colFlag=true;
                break;
            }
        }

        //看第一行是否有0
        for(int i=0;i<cols;++i)
        {
            if(matrix[0][i]==0)
            {
                rowFlag=true;
                break;
            }
        }

        //遍历其余矩阵元素，若有元素为0，则将对应的第一行第一列的位置置为0
        for(int i=1;i<rows;++i)
        {
            for(int j=1;j<cols;++j)
            {
                if(matrix[i][j]==0)
                {
                    matrix[i][0]=0;
                    matrix[0][j]=0;
                }
            }
        }

        //遍历其余元素并将标志为true的对应位置置为0
        for(int i=1;i<rows;++i)
        {
```

```

        for(int j=1;j<cols;++j)
        {
            if(matrix[i][0]==0 || matrix[0][j]==0)
            {
                matrix[i][j]=0;
            }
        }
    }

    //看第一列是否有0，有0则将该列置为0
    if(colFlag)
    {
        for(int i=0;i<rows;++i)
        {
            matrix[i][0]=0;
        }
    }

    //看第一行是否有0，有0则将该列置为0
    if(rowFlag)
    {
        for(int i=0;i<cols;++i)
        {
            matrix[0][i]=0;
        }
    }
}

};

```

AC

力扣
 学习
 题库
 讨论
 竞赛
 求职
 商店

下载App

Plus会员

中

🔔

🔔

📖 题目描述

💬 评论 (663)

🔑 题解 (924)

🕒 提交记录

执行结果: 通过 [显示详情](#)

➤ 添加备注

执行用时: 12 ms, 在所有 C++ 提交中击败了 83.03% 的用户
 内存消耗: 12.9 MB, 在所有 C++ 提交中击败了 35.14% 的用户
 通过测试用例: 164 / 164
 炫耀一下:

✍ 写题解, 分享我的解题思路

提交结果	执行用时	内存消耗	语言	提交时间	备注
通过	12 ms	12.9 MB	C++	2021/08/31 15:08	➤ 添加备注

① C++

• 智能模式

🕒

🔍

🔍

🔍

🔍

```

1  class Solution {
2  public:
3      void setZeroes(vector<vector<int>>& matrix)
4      {
5          if(matrix.size()==0 || matrix[0].size()==0) return;
6          bool rowFlag=false,colFlag=false; //两个flag记录第一
          行第一列是否有0
7          int rows=matrix.size();
8          int cols=matrix[0].size();
9
10         //看第一列是否有0
11         for(int i=0;i<rows;++i)
12         {
13             if(matrix[i][0]==0)
14             {
15                 colFlag=true;
16                 break;
17             }
18         }
19
20         //看第一行是否有0
21         for(int i=0;i<cols;++i)
22         {
23             if(matrix[0][i]==0)
24             {
25                 rowFlag=true;
26                 break;
27             }
28         }
29

```

测试用例

代码执行结果

调试器

