



AfterAcademy



OFFER

Android Online Course by MindOrks

Start your career in Android Development. Learn by doing real projects.

[ENROLL
NOW](#)

Admin AfterAcademy
14 Nov 2019

Set Matrix Zeros

Set Matrix Zeros - Interview Problem

AfterAcademy

Difficulty: Medium

Asked in: Amazon, Google

Understanding the Problem

Problem Description: Given a matrix, A of size M x N of 0's and 1's. If an element is 0, set its entire row and column to 0.

For Example:

Input:

```
[ [1, 0, 1],  
  [1, 1, 1],  
  [1, 1, 1] ]
```

Output:

```
[ [0, 0, 0],  
  [1, 0, 1],  
  [1, 0, 1] ]
```

Input:

```
[ [1, 0, 1],  
  [1, 1, 1],  
  [1, 0, 1] ]
```

Output:

```
[ [0, 0, 0],
```

[1, 0, 1],
[0, 0, 0]]

Possible follow-up questions to ask the interviewer:-

1. What are the constraints on the size of the matrix? (**Ans:** $1 \leq N, M \leq 1000$)
2. Do we need to change the rows and columns of elements which we set to zero? (**Ans:** No, only the zeroes present in the original matrix)

Solutions

We are discussing three ways to solve this problem:

1. **Brute Force Approach:** Using nested loops and extra space
2. **Using Hash Table:** Storing the status of rows and columns in the Hash Table.
3. **Using In-Place Hashing:** Storing hash in the matrix's first row and column.

1. Brute Force Approach

Common mistake

A simple solution that comes to mind for this problem is that we can traverse the matrix and for every '0' we encounter, we update all values in its row and column to 0. This brute force solution seems quite easy but may actually lead to a wrong answer. (**How? Think!**)

Correct Solution idea

One way would be to create an auxiliary matrix of the same size filled with 1's. We could make all changes required in this auxiliary matrix and then copy it to the original matrix in the end. We search for zeroes in the original matrix and update rows and columns in the auxiliary matrix.

Solution Steps

1. First, create a temporary matrix of the same size $M \times N$ and initialize its all places with 1.
2. Now scan the original matrix and if $A[i][j] == 0$ then set all the positions of row i and col j with 0 in new matrix.
3. Copy all elements from the temporary matrix to the original matrix.

Pseudo-Code

```
void setMatrixZeros(int A[][], int M, int N)
{
    int temp[M][N]
    for (i = 0 to M-1)
        for (j = 0 to N-1)
            temp[i][j] = 1
    for(i = 0 to M - 1)
    {
        for (j = 0 to N - 1)
        {
            if (A[i][j] == 0)
            {
                for (k = 0 to M - 1)
                    temp[k][j] = 0

                for (k = 0 to N - 1)
                    temp[i][k] = 0
            }
        }
    }
}
```

```
    }  
    for (i = 0 to M-1)  
        for (j = 0 to N-1)  
            arr[i][j] = temp[i][j]  
    }
```

Complexity Analysis

For every cell, we are traversing its row and column and there are a total of $N*M$ cells.

Time Complexity: $O(N*M*(N+M))$ (**Think!**)

Space Complexity: $O(N*M)$, for storing the auxiliary matrix.

Critical ideas to think!

- How could we implement it in place in order to optimize space complexity?
- How can we reduce the time complexity of the solution?

2. Using Hash Table

Instead of using an auxiliary matrix, we could just store the rows and columns which need to be updated (**Think!**).

We could use a Hash Table to do so. We shall create a hash table for all rows and columns and set them to false. If the value of any row or column in hash is later updated to true, it's a signal for us to update that particular row or column's value to 0.

Solution Steps

1. Create 2 Hash Tables **row** and **col** respectively of size M and N.
2. Now initialize all the values of row[] and col[] to **false**.
3. Now iterate over the matrix and for every $A[i][j] == 0$, set row[i] = true and col[j] = true.
4. After completion of step 3, again iterate through matrix A and for any $A[i][j]$, if row[i] or col[j] is true then update $A[i][j]$ to 0.

Pseudo Code

```
void setMatrixZeros(int A[][], int M, int N)
{
    bool row[M], col[N]

    for(i = 0 to M-1)
        row[i] = false
    for(j = 0 to N-1)
        col[j] = false

    for(i = 0 to M-1)
    {
        for(j = 0 to N-1)
        {
            if(A[i][j] == 0)
            {
                row[i] = true
                col[j] = true
            }
        }
    }

    for(i = 0 to M-1)
    {
        for(j = 0 to N-1)
```

```
        {  
            if(row[i] == true or col[j] == true)  
                A[i][j] = 0  
        }  
    }  
}
```

Complexity Analysis

Time Complexity: Creating and filling values in 2 Hash Tables + Traversing the matrix + Updating the matrix

$$= O(N+M) + O(N*M) + O(N*M)$$

$$= O(N*M)$$

Space Complexity: $O(M + N)$, for storing hash tables.

Critical ideas to think!

- How can we optimize the space complexity?
- What other matrix problems could be solved using a similar type of hashing approach?



NEW

Android App Development Online Course by MindOrks

Start your career in Android Development. Learn by doing real projects.

[CHECK NOW](#)

3. Using In-Place Hashing

We are using extra space to store Hash Tables of rows and columns. We can optimize this space by trying to store hash for row and column in the matrix itself. **(How?)**

We can use the first row and first column of the matrix to store the status of the rows and columns respectively. The problem will occur for the status of first row and column which can be handled using two variables which will store the status of the first row and column **(Think!)**

Solution Steps

1. Initialize **firstRow** and **firstCol** to false. These two variables will store the status of the first row and first column.
2. Now use the first row and first column as your hash which stores the status of that row and column.
3. Now iterate over the matrix and for every $A[i][j] == 0$, set $A[i][0] = 0$ and $col[0][j] = 0$.
4. Now update the values of the matrix except first row and first column to 0 if $A[i][0] = \text{true}$ or $A[0][i] = \text{true}$ for $A[i][j]$.
5. Now update the values of the first row and first column.

```
void setMatrixZeros(int A[][], int M, int N)
{
    for(j = 0 to N-1)
        if(A[0][j] == 0)
            firstRow = true

    for(i = 0 to M-1)
        if(A[i][0] == 0)
            firstCol = true

    for(i = 0 to M-1)
    {
        for(j = 0 to N-1)
        {
            if(A[i][j] == 0)
            {
                A[i][0] = 0
                A[0][j] = 0
            }
        }
    }

    for (i = 1 to M-1)
    {
        for (j = 1 to N-1)
```

```
        {
            if(A[0][j] == 0 || A[i][0] == 0)
                A[i][j] = 0
        }
    }

    if(firstRow == true)
    {
        for( i = 0 to N-1 )
            A[0][i] = 0
    }
    if(firstCol == true)
    {
        for(i = 0 to M-1)
            A[i][0] = 0
    }
}
```

Complexity Analysis

Time Complexity: Checking the first row and first column + Traversing and updating the matrix + Updating first row and first column.

$$= O(M) + O(N) + O(M*N) + O(M) + O(N)$$

$$= O(M*N)$$

Space Complexity: $O(1)$

Critical ideas to think!

- Why are the first row and first column updated separately?
- Can you use other data structures to efficiently solve this problem?

Comparison of different solutions

Approach	Time Complexity	Space Complexity
Brute Force: Using nested loops	$O(M*N*(M+N))$	$O(M*N)$
Using Hash Table	$O(M*N)$	$O(M+N)$
Using In-Place Hashing	$O(M*N)$	$O(1)$

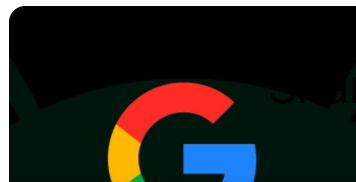
Suggested problems to solve

- Given a boolean matrix, find k such that all elements in the kth row are 0 and all elements in the kth column are 1
- Print unique rows in a given Boolean matrix
- Check if matrix can be converted to another matrix by transposing square sub-matrices
- Print a given matrix in spiral form
- Rotate a matrix by 90 degree

If you have any more approaches or you find an error/bug in the above solutions, please comment down below.

Happy Coding! Enjoy Algorithms!

AfterAcademy Data Structure And Algorithms Online Course - Admissions Open



NEW

Share this blog and spread the knowledge
Google Android Developer

SHARE ON FACEBOOK

SHARE ON TWITTER

SHARE ON LINKEDIN

SHARE ON TELEGRAM

SHARE ON REDDIT

SHARE ON WHATSAPP

Recommended for you

Interview question

LRU Cache implementation



Asked in

afteracademy.com

LRU Cache Implementation

Design and implement a data structure for Least Recently Used(LRU) cache. Your data structure must support two operations: get(key) and put(). The problem expects a constant time solution



Admin AfterAcademy
12 Oct 2020

Interview question

Letter Combinations of a Phone Number



Asked in

afteracademy.com

Letter Combinations of a Phone Number

Given a string str, containing digits from 2 - 9 inclusive, write a program to return all the possible letter combinations that the number could represent. This is a popular coding interview question based on backtracking and recursive approach.



Admin AfterAcademy
11 Oct 2020

Interview question

Interleaving String

ABMN
AMBN
AMNB



Admin AfterAcademy
17 Jul 2020



Admin AfterAcademy
5 Oct 2020

Reverse a Stack Using Recursion

Reverse a Stack Using Recursion

afteracademy.com

Reverse a Stack Using Recursion

Given a stack of integers `st`, write a program to reverse the stack using recursion. This problem will clear the concept of recursion.

Asked in  

afteracademy.com

Interleaving Strings

Given three strings `S1`, `S2` and `S3`, write a program which checks whether `S3` is an interleaving of `S1` and `S2`. The problem is a typical dynamic programming problem.


Interview question

Surrounded regions

```

X X O O X O
O X O X X X

```

Asked in 

afteracademy.com

Surrounded regions

Given a 2-D matrix board where every element is either 'O' or 'X', write a program to replace 'O' with 'X' if surrounded by 'X'. It is a famous problem based on the concept of flood fill algorithms



Admin AfterAcademy
22 Sep 2020



Recursive Insertion Sort

afteracademy.com

Recursive Insertion Sort

Write a program for the recursive implementation of Insertion Sort. Insertion Sort is used to sort a given array. This problem will sharpen your recursion skills.



Admin AfterAcademy
22 Sep 2020

Our Learners Work At



AfterAcademy

Stay up to date. Follow us on



© Copyright 2019

MindOrks Nextgen Private Limited
Gurgaon, Haryana, India
+91-8287460223

Quick Links

[Contact Us](#)
[Privacy Policy](#)
[Terms And Conditions](#)
[Cookie Policy](#)

About Us

MindOrks
Amit Shekhar
Janishar Ali

Free Resources

[Publication](#)
[Medium](#)
[Video Lessons](#)
[Open Source](#)