

Print unique rows in a given boolean matrix

Given a binary matrix, print all unique rows of the given matrix.

Input :

```
{0, 1, 0, 0, 1}
{1, 0, 1, 1, 0}
{0, 1, 0, 0, 1}
{1, 1, 1, 0, 0}
```

Output :

```
0 1 0 0 1
1 0 1 1 0
1 1 1 0 0
```

Method 1 (Simple)

A simple approach is to check each row with all processed rows. Print the first row. Now, starting from the second row, for each row, compare the row with already processed rows. If the row matches with any of the processed rows, don't print it. If the current row doesn't match with any row, print it.

Time complexity: $O(\text{ROW}^2 \times \text{COL})$

Auxiliary Space: $O(1)$

Method 2 (Use Binary Search Tree)

Find the decimal equivalent of each row and insert it into BST. Each node of the BST will contain two fields, one field for the decimal value, other for row number. Do not insert a node if it is duplicated. Finally, traverse the BST and print the corresponding rows.

Time complexity: $O(\text{ROW} \times \text{COL} + \text{ROW} \times \log(\text{ROW}))$

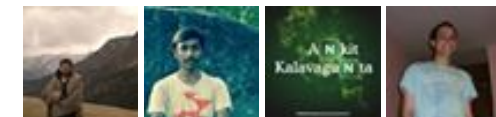
Google™ Custom Search



GeeksforGeeks



53,520 people like [GeeksforGeeks](#).



[Interview Experiences](#)

[Advanced Data Structures](#)

[Dynamic Programming](#)

[Greedy Algorithms](#)

[Backtracking](#)

[Pattern Searching](#)

[Divide & Conquer](#)

[Mathematical Algorithms](#)

[Recursion](#)

[Geometric Algorithms](#)

Auxiliary Space: $O(\text{ROW})$

This method will lead to Integer Overflow if number of columns is large.

Method 3 (Use Trie data structure)

Since the matrix is boolean, a variant of Trie data structure can be used where each node will be having two children one for 0 and other for 1. Insert each row in the Trie. If the row is already there, don't print the row. If row is not there in Trie, insert it in Trie and print it.

Below is C implementation of method 3.

```
// Given a binary matrix of M X N of integers, you need to return only
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

#define ROW 4
#define COL 5

// A Trie node
typedef struct Node
{
    bool isEndOfCol;
    struct Node *child[2]; // Only two children needed for 0 and 1
} Node;

// A utility function to allocate memory for a new Trie node
Node* newNode()
{
    Node* temp = (Node *) malloc( sizeof( Node ) );
    temp->isEndOfCol = 0;
    temp->child[0] = temp->child[1] = NULL;
    return temp;
}

// Inserts a new matrix row to Trie. If row is already
// present, then returns 0, otherwise inserts the row and
// return 1
bool insert( Node** root, int (*M)[COL], int row, int col )
{
    // base case
    if ( *root == NULL )
        *root = newNode();
```



Popular Posts

[All permutations of a given string](#)

[Memory Layout of C Programs](#)

[Understanding "extern" keyword in C](#)

[Median of two sorted arrays](#)

[Tree traversal without recursion and without stack!](#)

[Structure Member Alignment, Padding and Data Packing](#)

[Intersection point of two Linked Lists](#)

[Lowest Common Ancestor in a BST.](#)

[Check if a binary tree is BST or not](#)

[Sorted Linked List to Balanced BST](#)

```

// Recur if there are more entries in this row
if ( col < COL )
    return insert ( &(amp; (*root)->child[ M[row][col] ] ), M, row, col );

else // If all entries of this row are processed
{
    // unique row found, return 1
    if ( !( (*root)->isEndOfCol ) )
        return (*root)->isEndOfCol = 1;

    // duplicate row found, return 0
    return 0;
}

// A utility function to print a row
void printRow( int (*M)[COL], int row )
{
    int i;
    for( i = 0; i < COL; ++i )
        printf( "%d ", M[row][i] );
    printf("\n");
}

// The main function that prints all unique rows in a
// given matrix.
void findUniqueRows( int (*M)[COL] )
{
    Node* root = NULL; // create an empty Trie
    int i;

    // Iterate through all rows
    for ( i = 0; i < ROW; ++i )
        // insert row to TRIE
        if ( insert(&root, M, i, 0) )
            // unique row found, print it
            printRow( M, i );
}

// Driver program to test above functions
int main()
{
    int M[ROW][COL] = {{0, 1, 0, 0, 1},
                        {1, 0, 1, 1, 0},
                        {0, 1, 0, 0, 1},
                        {1, 0, 1, 0, 0}};
};

```

Deploy Early. Deploy Often.

DevOps from
Rackspace:

Automation

[FIND OUT HOW ►](#)



```

findUniqueRows ( M );

return 0;
}

```

Time complexity: $O(\text{ROW} \times \text{COL})$

Auxiliary Space: $O(\text{ROW} \times \text{COL})$

This method has better time complexity. Also, relative order of rows is maintained while printing.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

Find + Remove Duplicates

 dataladder.com/RemoveDuplicates

\$0 License and free support Clean up your duplicate data today!



Recent Comments

Aman Hi, Why arent we checking for conditions...

[Write a C program to Delete a Tree.](#) · 6 minutes ago
kzs please provide solution for the problem...

[Backtracking | Set 2 \(Rat in a Maze\)](#) · 10 minutes ago

Sanjay Agarwal bool

[tree::Root_to_leaf_path_given_sum\(tree...](#)
Root to leaf path sum equal to a given number · 35 minutes ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

[Count trailing zeroes in factorial of a number](#) · 37 minutes ago

newCoder3006 If the array contains negative numbers also. We...

[Find subarray with given sum](#) · 1 hour ago

newCoder3006 Code without using while loop. We can do it...


[Find subarray with given sum](#) · 1 hour ago

AdChoices 

[▶ Matrix in Java](#)

[▶ Rows N Columns](#)

[▶ Rows](#)

AdChoices 

Related Tpoics:

- [Remove minimum elements from either side such that 2*min becomes more than max](#)
- [Divide and Conquer | Set 6 \(Search in a Row-wise and Column-wise Sorted 2D Array\)](#)
- [Bucket Sort](#)
- [Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1](#)
- [Find the number of zeroes](#)

- Find if there is a subarray with 0 sum
- Divide and Conquer | Set 5 (Strassen's Matrix Multiplication)
- Count all possible groups of size 2 or 3 that have sum as multiple of 3



1



Tweet

0



0

[▶ Matrix Code](#)

[▶ Matrix Type](#)

[▶ Compare Matrix](#)

AdChoices

[▶ Find Matrix](#)

[▶ What Is Matrix](#)

[▶ Can Matrix](#)

Writing code in comment? Please use ideone.com and share the link here.

@geeksforgeeks, **Some rights reserved**

Contact Us!

Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team