

Turn an image by 90 degree

Given an image, how will you turn it by 90 degrees? A vague question. Minimize the browser and try your solution before going further.

An image can be treated as 2D matrix which can be stored in a buffer. We are provided with matrix dimensions and it's base address. How can we turn it?

For example see the below picture,

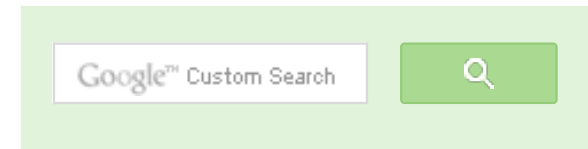
```
* * * ^ * * *
* * * | * * *
* * * | * * *
* * * | * * *
```

After rotating right, it appears (observe arrow direction)

```
* * * *
* * * *
* * * *
- - - - >
* * * *
* * * *
* * * *
```

The idea is simple. Transform each row of source matrix into required column of final image. We will use an auxiliary buffer to transform the image.

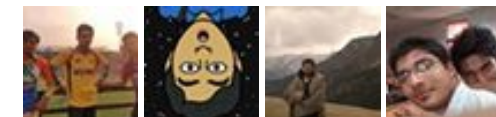
From the above picture, we can observe that



GeeksforGeeks



53,522 people like [GeeksforGeeks](#).



[Interview Experiences](#)

[Advanced Data Structures](#)

[Dynamic Programming](#)

[Greedy Algorithms](#)

[Backtracking](#)

[Pattern Searching](#)

[Divide & Conquer](#)

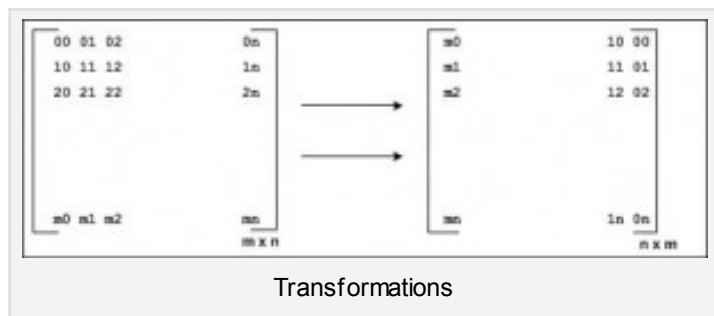
[Mathematical Algorithms](#)

[Recursion](#)

[Geometric Algorithms](#)

first row of source -----> last column of destination
 second row of source -----> last but-one column of destination
 so ... on
 last row of source -----> first column of destination

In pictorial form, we can represent the above transformations of an (m x n) matrix into (n x m) matrix,



If you have not attempted, atleast try your pseudo code now.

It will be easy to write our pseudo code. In C/C++ we will usually traverse matrix on row major order. Each row is transformed into different column of final image. We need to construct columns of final image. See the following algorithm (transformation)

```
for(r = 0; r < m; r++)
{
    for(c = 0; c < n; c++)
    {
        // Hint: Map each source element indices into
        // indices of destination matrix element.
        dest_buffer [ c ] [ m - r - 1 ] = source_buffer [ r ] [ c ];
    }
}
```

Note that there are various ways to implement the algorithm based on traversal of matrix, row major or column major order. We have two matrices and two ways (row and column major) to traverse each matrix. Hence, there can atleast be 4 different ways of transformation of source matrix into final matrix.

Code:

Take the First Step
LEARN MORE ABOUT ITT TECH

ITT Educational Services, Inc. **ITT** >>> **CLICK HERE**

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](#)

Code:

```
#include <stdio.h>
#include <stdlib.h>

void displayMatrix(unsigned int const *p, unsigned int row, unsigned int col)
void rotate(unsigned int *pS, unsigned int *pD, unsigned int row, unsigned int col)

int main()
{
    // declarations
    unsigned int image[][4] = {{1,2,3,4}, {5,6,7,8}, {9,10,11,12}};
    unsigned int *pSource;
    unsigned int *pDestination;
    unsigned int m, n;

    // setting initial values and memory allocation
    m = 3, n = 4, pSource = (unsigned int *)image;
    pDestination = (unsigned int *)malloc(sizeof(int)*m*n);

    // process each buffer
    displayMatrix(pSource, m, n);

    rotate(pSource, pDestination, m, n);

    displayMatrix(pDestination, n, m);

    free(pDestination);

    getchar();
    return 0;
}

void displayMatrix(unsigned int const *p, unsigned int r, unsigned int c)
{
    unsigned int row, col;
    printf("\n\n");

    for(row = 0; row < r; row++)
    {
        for(col = 0; col < c; col++)
        {
            printf("%d\t", *(p + row * c + col));
        }
        printf("\n");
    }
}
```



LEARN MORE

EXPLORE YOUR
OPTIONS
AT **ITT TECH**

ITT Technical Institute **ITT**

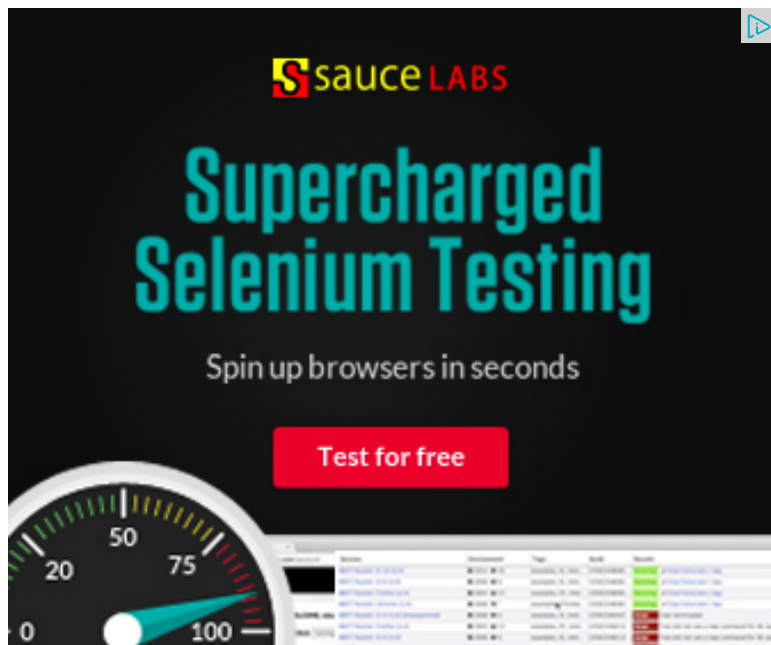
```

    printf("\n\n");
}

void rotate(unsigned int *pS, unsigned int *pD, unsigned int row, unsigned int col)
{
    unsigned int r, c;
    for(r = 0; r < row; r++)
    {
        for(c = 0; c < col; c++)
        {
            *(pD + c * row + (row - r - 1)) = *(pS + r * col + c);
        }
    }
}

```

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



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](#)

Recent Comments

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

[Write a C program to Delete a Tree](#) · 26 minutes ago

kzs please provide solution for the problem...

[Backtracking | Set 2 \(Rat in a Maze\)](#) · 30 minutes ago

Sanjay Agarwal bool

tree::Root_to_leaf_path_given_sum(tree...
Root to leaf path sum equal to a given number · 55 minutes ago

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

[Count trailing zeroes in factorial of a number](#) · 57 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 

[► C++ Vector](#)

[► Matrix in Java](#)

- 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



0



Tweet

0

1

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

32 Comments

GeeksforGeeks

Sort by Newest ▼



Join the discussion...



alien • 3 months ago

most easiest algorithm

```
int i,j,row=4,col=0;
```

```
for(i=0;i<3;i++,col++)
```

```
{
```

```
row=4;
```

```
for(j=0;j<5;j++,row--)
```

```
{
```

```
arr2[i][j] = arr[row][col];
```

```
}
```

AdChoices

► [Rotate a Image](#)

► [Programming C++](#)

► [Image String](#)

AdChoices

► [Flip Image](#)

► [Vector an Image](#)

► [Image Source](#)

J

```
for(i=0;i<3;i++)
```

see more

^ | v .



nidhi · 8 months ago

How to do in-place?

^ | v .



anonymous · 10 months ago

<http://stackoverflow.com/questions/2893101/how-to-rotate-a-n-x-n-mat>

^ | v .



wasseypuriyan → anonymous · 7 months ago

Do we any solution for general NxM in place transformation

1 ^ | v .



its_dark → wasseypuriyan · a month ago

```
for(int i=0; i < n / 2 ; i++)
    for(int j=0; j<(n+1)/2; j++)
        cyclic_roll(m[i][j], m[n-1-j][i], m[n-1-i][n-1-j], n
```

```
void cyclic_roll(int &a, int &b, int &c, int &d)
{
    int temp = a;
    a = b;
```

```
    d = c;  
    c = d;  
    d = temp;  
}
```

^ | v .



shivi · 10 months ago

//shivi..coding is addictive!!

```
#include<shiviheaders.h>
```

```
#define M 4
```

```
#define N 7
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    char brr[N][M];
```

```
    char arr[][N]=
```

```
{
```

```
{ '*', '*', '*', '^', '*', '*', '*' }, { '*', '*', '*', '|', '*', '*', '*' }, {
```

```
};
```

```
int x=N-1,y=M-1;
```

```
for(int i=0;i<M;++i)
```

```
{
```

see more

^ | v .



will smith · 11 months ago



```
/ Paste your code here (you may delete these lines if not writing C)
#include<stdio.h>

int main()
{
    int n,m,i,j;
    scanf("%d%d",&n,&m);
    int src[n][m],des[m][n];
    for(i=0;i<n;i++)
        for(j=0;j<m;j++)
            scanf("%d",&src[i][j]);
    printf("Initial Matrix\n");
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++)
        {
            printf("%2d ",src[i][j]);
        }
        printf("\n");
    }
}
```

[see more](#)

^ | v •



Sreenivas Doosa • a year ago

First transpose the matrix and then swap the columns. Here is the working co

[sourcecode language="java"]

```
package com.ds.algo;
```

```
public class Matrix {
```

```
    int input[];
```

```
    int ROWS, COLS;
```



```
public static void main(String args[]) {  
  
    int M = 4;  
    int N = 3;  
    Matrix matrix = new Matrix(M, N);  
    System.out.println("Original Matrix -->");  
    matrix.printMatrix(M, N);  
    matrix.rotateBy90();  
    System.out.println("After rotating by 90 -->");  
}
```

[see more](#)

^ | v ·



Hanish · a year ago

"We have two matrices and two ways (row and column major) to traverse each different way of transformation of source matrix into final matrix."

Can you please explain this ??

We will traverse either source matrix in row or column major and fill the destination not be just 2 ways?

Please elaborate the 4 different PSEUDO procedures of traversal.

^ | v ·



Kunal Chitkara · 2 years ago

Another very simple method without using any pointers

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
    int image[][4] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 9, 9, 9};
```

```
    int i, j, rows, cols, i1, j1;
```

```
    rows = sizeof(image) / sizeof(image[0]);
```

```
cols = sizeof(image[0])/sizeof(image[0][0]);
```

```
int rot[cols][rows] ;  
printf("Original Matrix:\n");  
for(i = 0; i < rows; i++)  
{  
    for(j = 0; j < cols; j++)  
        printf("%d ", image[i][j]);  
    printf("\n");  
}
```

[see more](#)

^ | v .



vivek · 2 years ago

is it not taking transpose of a matrix ?

^ | v .



algopiggy → **vivek** · a year ago

In transpose, the first row will become the first column. Here the first row of the matrix after the transformation. :)

```
/* Paste your code here (You may delete these lines if not writ
```

^ | v .



Diksha · 2 years ago

I have tried doing it inplace & its working.
here is the code.

```
int main()
```

```

{
    int old[][3]={1,2,3,4,5,6,7,8,9},i,first,last,j,L,offset;
    clrscr();

    for(L=0;L<3/2;L++)
    {
        first=L;
        last=3-1-L;
        for(i=0;i<last;i++)
        {
            offset=i-first;
            j=old[first][i];

```

see more

^ | v .



Bhupendra → Diksha · 2 years ago

This will work only for case where number of rows and column are equal. It is straight forward. You have to deal with cycles. see wiki for in place transposition

```

/* Paste your code here (You may delete these lines if not writing)

```

^ | v .



jIntrnal frame · 2 years ago

Thanks for finally talking about >Turn an image by 90 degree | GeeksforGeeks
jPanel frame into jDesktopPane

^ | v .



Priyanka · 2 years ago



```
#define ROW 2
#define COL 3
//Rotating an image by 90 degree
void rotateimage(int a[][COL])
{
    int i,j;
    printf("\nRotated Image\n");
    for(i=0;i<COL;i++)
    {
        for(j=ROW-1;j>=0;j--)
            printf("%d ",a[j][i]);
        printf("\n");
    }
}
```

^ | v .



Priya T B · 3 years ago

Let $a[n][n]$ be the pixel matrix.

```
for(int i=0;i<n/2;i++){
    for(int j=0;j<n+1/2;j++){
        exchange(a[i][j],a[n-j-1][i],a[n-i-1][n-j-1],a[j][n-1-i]);
    }
}
```

```
exchange(int &a, int &b, int &c, int &d){
    int temp;
    temp = a;
```

```
a=b,  
b=c;  
c=d;  
d=temp;  
}
```

^ | v .



Priya T B · 3 years ago

//Let a[][] be n*n the 2d array having the pixel values...

```
for(int i=0;i<n/2;i++){  
    for(int j=0;j<n+1/2;j++){  
        exchange(a[i][j],a[n-j-1][i],a[n-i-1][n-j-1],a[j][n-1-i]);  
    }  
}
```

```
exchange(int &a, int &b, int &c, int &d){  
    int temp;  
    temp = a;  
    a=b;  
    b=c;  
    c=d;  
    d=temp;  
}
```

^ | v .



bond · 3 years ago

I think in place rotation can be done by first taking transpose of matrix and ther

with last, second with second last and so on..)
complexity- $O(n^2)$

^ | v .



wgpshashank · 3 years ago

In Place Matrix Rotation

```
void rotate_matrix_90degree(int **m, int n)
{
    int i, j;

    // first mirror the matrix along the diagonal line.
    for (i = 0; i < n; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            int tmp = m[i][j];
            m[i][j] = m[j][i];
            m[j][i] = tmp;
        }
    }
}
```

[see more](#)

^ | v .



extremecode → wgpshashank · 2 years ago

@wgpshashank your first loop has no effect on the matrix, simulate and

/* Paste your code here (You may delete these lines if not writ

^ | v .



Algoseekar → wgpshashank · 3 years ago

@wgpshashank..yes it will work only for square matrix & fail for $m \times n$ m to solve it..inplace..isn't it..??

^ | v .



abc → wgpshashank · 3 years ago

But this will work only for square matrix. Right?

^ | v .



Sajeesh · 3 years ago

```
//for each point-(0,0)(1,1) and so on...
for(i=0;i<rows;i++)
{
    //traversing along the row and col from the point(i,i) and swap
    t=i+1;
    while(t<rows||t<cols){
        swap(a[i][t],a[t][i]);
        t++;
    }
}
```

^ | v .



Gaurav · 3 years ago

I think we can implement it in-place by doing a four way swap by first starting from elements clockwise from the edges. I hope you understood what i am trying to

^ | v .



Jing → Gaurav · 3 years ago

I agree

^ | v ·



harry → Jing · 3 years ago

Can somebody please provide code/algorithm for in-place rotat

^ | v ·



sharat → harry · 3 years ago

check this link

<http://stackoverflow.com/quest...>

for some interesting ideas on inplace rotation

^ | v ·



kartik · 3 years ago

nice in-place implementation :)

^ | v ·



Algoseekar → kartik · 3 years ago

@kartik...wgpshashank has provided inplace implementation for squar
for general $m \times n$ matrix..???

^ | v ·



Venki → kartik · 3 years ago

No, it is not in-place. I am using $O(mn)$ extra buffer for transformations

^ | v ·



sharat → Venki · 3 years ago

probably Karthik was sarcastic :)

^ | v ·



Subscribe



Add Disqus to your site

@geeksforgeeks, **Some rights reserved**

Contact Us!

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