Lesson:

2D Array Problems Set 3

Pre-Requisites

- Concept of Arrays and vectors
- Basic understanding of multidimensional arrays
- 2D Vectors Concept

List of Concepts Involved

- Row with the maximum number of 1s.
- Rotation of matrix

Problems

Q1. Given a boolean 2D array, where each row is sorted. Find the row with the maximum number of 1s.

Example 1:

Input:
row = 3, col = 4
matrix[ ] = {{0 1 1 1},
{0 0 0 1},
{0 0 0 1}}

Output: 0

Explanation: Row 0 has 3 ones whereas rows 1 and 2 have just 1 ones.

Solution approach:

- A simple approach is to do a row-wise traversal of the 2D array.
- Count the number of 1s in each row, and compare the count with max.
- Finally, return the index of the row with maximum 1s.

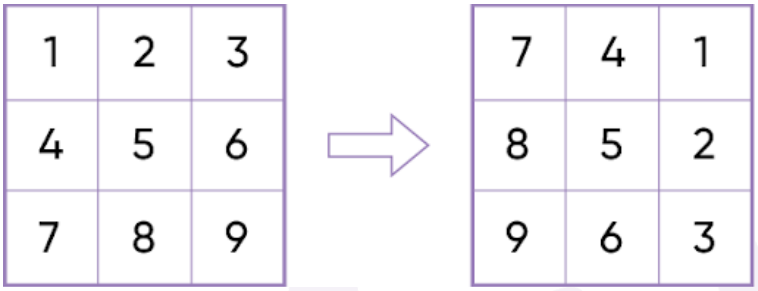
Let us now write the implementation code.

```
#include <iostream>
#include<vector>
using namespace std;
int max_1s(vector<vector<int>>matrix)
{
    int max = 0; // maximum number of 1's
    int max_row = 0; // index of the row having the maximum number of 1's
    for (int i = 0; i < matrix.size(); i++)
    {
        int count = 0; // number of 1's in current row
        for (int j = 0; j < matrix[0].size(); j++)
        {
            if (matrix[i][j] == 1)
                count++;
        }
        if (count > max)
        { max = count;
          max_row = i;
        }
    }
    return max_row;
}
int main()
{ int n,m;
  cout<<"enter the number of rows you want : ";
  cin>>n;
  cout<<"enter the number of columns you want : ";
  cin>>m;
  vector<vector<int>> matrix(n,vector<int>(m));
  cout<<"enter the elements of the matrix : ";
  for(int i=0;i<n;i++){
    for(int j=0;j<m;j++){
        cin>>matrix[i][j];
    }
  }
  int max = max_1s(matrix);
  cout << "The row with maximum ones is : "<<max;
  return 0;
}
```

OUTPUT :

```
enter the number of rows you want : 3
enter the number of columns you want : 3
enter the elements of the matrix :
0 0 1
1 1 1
1 0 1
The row with maximum ones is : 1
```

Q2: Given a square matrix, turn it by 90 degrees in a clockwise direction without using any extra space.



Approach
An N x N matrix will have floor(N/2) square cycles.

For example, a 3 X 3 matrix will contain 1 cycle. The cycle is formed by its 1st row, last column, last row, and 1st column.

For each square cycle, elements are swapped and involved with the corresponding cell in the matrix in the clockwise direction. We require a temporary variable for this.

Let us get down to coding this.

```
#include<iostream>
#include<vector>
using namespace std;
 void rotate(vector<vector<int>>& matrix) {
        int N=matrix[0].size();
        for (int i = 0; i < N / 2; i++) {
        for (int j = i; j < N - i - 1; j++) {
            // Swap elements of each cycle
            // in clockwise direction
            int temp = matrix[i][j];
            matrix[i][j] = matrix[N - 1 - j][i];
            matrix[N - 1 - j][i] = matrix[N - 1 - i][N - 1 - j];
            matrix[N - 1 - i][N - 1 - j] = matrix[j][N - 1 - i];
            matrix[j][N - 1 - i] = temp;
        }
    }
}
int main(){
    int row , col;
   cout<<"Enter number of rows : ";
   cin>>row;
   cout<<"Enter number of columns :";
   cin>>col;
   vector<vector<int>>arr(row , vector<int>(col));
   for(int i=0;i<row;i++){
       for (int j=0;j<col;j++){
           cin>>arr[i][j];
       }
   }
   rotate(arr);
   cout<<"The rotated matrix is as shown below : "<<endl;
    for(int i=0;i<row;i++){
       for (int j=0;j<col;j++){
           cout<<arr[i][j]<<" ";
       }
       cout<<endl;
   }
}
```

OUTPUT :

```
Enter number of rows : 3
Enter number of columns :3
4 5 6
2 5 3
1 2 7
The rotated matrix is as shown below :
1  2  4
2  5  5
7  3  6
```

Upcoming Class Teasers

- 2D Array advance problems