https://course.acciojob.com/idle?question=40ad500f-e244-4123-96fa-fb6300ce8eac

- EASY
- Max Score: 30 Points
-

Row with Maximum 1's

You have been given a non-empty grid `MAT` with `N` rows and `M` columns consisting of only 0s and 1s. All the rows are sorted in ascending order.

Your task is to find the index of the row that has the maximum number of ones.

Note: If two rows have the same number of ones, consider the one with a smaller index.

Input Format

The first input line contains two space separated integers `N` and `M` representing the number of rows and columns of the grid, respectively.

From the second line, the next `N` lines represent the rows of the grid. Every row contains `M` single space-separated integers.

Output Format

Print the index of the row with the maximum number of ones.

Example 1

Input

```
2 2
1 1
1 1
```

Output:

```
0
```

Explanation For the given grid, both rows have an equal number of ones. Since the row with index 0 has a smaller index. Hence the answer is 0.

Example 2

Input

```
3 4
0 0 1 1
0 0 0 1
0 1 1 1
```

Output:

```
2
```

Explanation For the given grid, third row has highest number of 1s. So, output is 2 (0 based index).

Constraints

1 <= `N` <= 500

1 <= `M` <= 500

0 <= `MAT[i][j]` <= 1

Topic Tags

- **2D-Arrays**
- **Binary Search**

My code

// in java

```
import java.util.*;
import java.lang.*;
import java.io.*;

public class Main
{
        public static void main (String[] args) throws java.lang.Exception
        {
                //your code here
      Scanner s=new Scanner(System.in);
      int n=s.nextInt();
            int m=s.nextInt();
      int arr[][]=new int[n][m];
      for(int i=0;i<n;i++)
        for(int j=0;j<m;j++)
          arr[i][j]=s.nextInt();


      int max=0,ind=0;
       for(int i=0;i<n;i++)
       {
                  int c=0;
          for(int j=0;j<m;j++)
          {
            if(arr[i][j]==1) c++;
          }
          if(c>max)
                  {
                          max=c;
                          ind=i;
                  }
       }
```

```
        System.out.print(ind);
          }


}
```