Jadoo and Face Detection - Hackerearth

Monday, March 9, 2020 9:39 PM

The Problem:

Jadoo has become quite interested in Computer Vision lately, especially in the detection of faces. He became interested in Viola-Jones algorithm.

In Viola-Jones algorithm, the input image is first reduced to grayscale to reduce the amount of information and then a sliding window moves across the image and tries to match the information in the window along with previously given sets of facial and non-facial images. One of the things the Viola-Jones algorithm compares is the total intensity in the sliding window.

Let us consider the given image to be a matrix of r rows and c columns where image[i][j] represents the intensity of a single pixel. Given a set of q queries in the form of x1, y1 and x3, y3 signifying the opposite corners of the sliding window, find the total intensity of a window.

Input Format: The first line contains two integers r and c. The next r lines contain c integers each (signifying the image as a matrix). The next lines contain an integer q. The next q lines consist of x1, y1, x2 and y2 each.
Output Format: The output should contain q lines signifying the total intensity in each window of the image.
Please Note that the matrix is 0 indexed

SAMPLE INPUT

```
8 10
1 2 5 7 2 8 0 6 4 6
9 8 0 4 9 5 10 7 10 3
7 6 10 2 0 10 4 9 10 8
3 8 1 5 4 8 0 9 5 8
9 5 0 1 3 4 1 9 6 1
1 2 5 6 9 9 0 2 4 0
1 2 4 1 6 6 10 4 2 5
5 6 2 10 5 3 9 10 10 2
1
2 5 5 7
```

SAMPLE OUTPUT

```
65
```

Explanation

The submatrix is :
10 4 9

```
8 0 9
4 1 9
9 0 2
```

Thus, the sum is `10 + 4 + 9 + 8 + 0 + 9 + 4 + 1 + 9 + 9 + 0 + 2 = 65`

The Code:

```
/* While taking the input: for every index in a row, store the sum of all the row elements until that index
(including it).
*Then sum of a range of elements between indices i and j (j > i) in that row will simply be arr[j] - arr[i-1].
*/

#include <stdio.h>

int main(){
        int row, col, pixel, caseCount, x1, y1, x2, y2, i, j;
        int leftBound, rightBound, upperBound, lowerBound, sum;

        scanf("%d %d", &row, &col);

        int** image = (int**)malloc(row * sizeof(int*));

        for(i=0; i<row; i++) {
           image[i] = (int*)malloc((col+1) * sizeof(int));
           image[i][0] = 0;
           for(j=1; j<=col; j++) {
              scanf("%d", &pixel);
              image[i][j] = image[i][j-1] + pixel;
           }
        }

        scanf("%d", &caseCount);

        while(caseCount > 0) {
           scanf("%d %d %d %d", &x1, &y1, &x2, &y2);

           sum = y1 + y2;
           leftBound = y1 > y2 ? y2 : y1;
           rightBound = sum - leftBound;

           sum = x1 + x2;
           upperBound = x1 > x2 ? x2 : x1;
           lowerBound = sum - upperBound;

           sum = 0;
           for(i=upperBound; i<=lowerBound; i++) {
              sum += (image[i][rightBound+1] - image[i][leftBound]);
           }

           printf("%d\n", sum);

           caseCount--;
        }

        free(image);
}
```

The Stats:

Score

30.0

Time (sec)

0.20254

Memory (KiB)

64

Language

C