**算法训练 暗恋**

时间限制：1.0s   内存限制：256.0MB

问题描述

　　同在一个高中，他却不敢去找她，虽然在别人看来，那是再简单不过的事。暗恋，是他唯一能做的事。他只能在每天课间操的时候，望望她的位置，看看她倾心的动作，就够了。操场上的彩砖啊，你们的位置，就是他们能够站立的地方，他俩的关系就像砖与砖之间一样固定，无法动摇。还记得当初铺砖的工人，将整个操场按正方形铺砖（整个操场可视为R行C列的矩阵，矩阵的每个元素为一块正方形砖块），正方形砖块有两种，一种为蓝色，另一种为红色。我们定义他和她之间的“爱情指标”为最大纯色正方形的面积，请你写一个程序求出“爱情指标”。

输入格式

　　第一行两个正整数R和C。  
　　接下来R行C列描述整个操场，红色砖块用1来表示，蓝色砖块用0来表示。

输出格式

　　一个数，表示他和她之间的“爱情指标”。

样例输入

5 8  
0 0 0 1 1 1 0 1  
1 1 0 1 1 1 1 1  
0 1 1 1 1 1 0 1  
1 0 1 1 1 1 1 0  
1 1 1 0 1 1 0 1

样例输出

9

数据规模和约定

　　40%的数据R,C<=10;  
　　70%的数据R,C<=50;  
　　100%的数据R,C<=200;

本题的C++参考代码如下：

#include<cstdio>

#include<iostream>

#include<algorithm>

using namespace std;

int r,c;

int area(int a[200][200],int i,int j,int n)

{

int k,l;

if(n==min(r-1,c-1))

return n;

else

{

for(k=0;k<=n;k++)

{

if(a[i+n][j+k]!=a[i][j])

{

return n;

}

}

for(l=0;l<n;l++)

{

if(a[i+l][j+n]!=a[i][j])

return n;

}

area(a,i,j,++n);

}

}

int main()

{

// freopen("input3(1).txt","r",stdin);

int i,j,max=0,n=1,s;

scanf("%d%d",&r,&c);

int a[200][200]={-1};

// memset(a,-1,sizeof(a));

for(i=0;i<r;i++)

{

for(j=0;j<c;j++)

scanf("%d",&a[i][j]);

}

// for(i=0;i<r;i++)

// {

// for(j=0;j<c;j++)

// printf("%d ",a[i][j]);

// printf("\n");

// }

for(i=0;i<r-1;i++)

{

for(j=0;j<c-1;j++)

{

n=area(a,i,j,1);

if(n>max) max=n;

// printf("a[%d][%d]=%d\n",i,j,n);

if(max>=c/2)

break;

}

if(max>=r/2)

break;

}

printf("%d",max\*max);

return 0;

}

本题的C参考代码如下：

#include <stdio.h>

#define MAXSIZE 222

int R, C, a[MAXSIZE][MAXSIZE];

void newLoveIndex(int x, int y, int \*sq)

{

int i, j, color, s, t;

if(x+(\*sq) > R || y+(\*sq) > C){

return;

}

color = a[x][y];

for(i=x; i< x+(\*sq); i++){

for(j=y; j< y+(\*sq); j++){

if(a[i][j] != color){

return;

}

}

}

s = i, t = j;

while(a[s][t] == color && s < R && t < C){

i = s, j = t;

while(--i && i>= x){

if(a[i][t] != color){

return;

}

}

while(--j && j>= y){

if(a[s][j] != color){

return;

}

}

(\*sq) ++, s ++, t ++;

}

}

int main()

{

int i, j, k, sqrtLI, loveIndex;

scanf("%d%d", &R, &C);

for(i=0; i<R; i++){

for(j=0; j<C; j++){

scanf("%d", &a[i][j]);

}

}

sqrtLI = 1;

for(i=0; i<R; i++){

for(j=0; j<C; j++){

newLoveIndex(i, j, &sqrtLI);

}

}

loveIndex = sqrtLI \* sqrtLI;

printf("%d", loveIndex);

return 0;

}

本题的Java参考代码如下：

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.StreamTokenizer;

public class Main {

private static int x, y;

private static int arr[][] = new int[205][205];

private static int max = 0, p, q;

public static void main(String[] args) throws IOException {

StreamTokenizer st = new StreamTokenizer(new BufferedReader(new InputStreamReader(System.in)));

st.nextToken();

x = (int) st.nval;

st.nextToken();

y = (int) st.nval;

for (int i = 1; i <= x; i++) {

for (int j = 1; j <= y; j++) {

st.nextToken();

arr[i][j] = (int) st.nval;

}

}

for (int i = 1; i <= x; i++) {

for (int j = 1; j <= y; j++) {

baosou(i, j, arr[1][i], 1);

}

}

boolean is = false;

q = q - max;

for (int i = p; i > p - max; i--) {

if (arr[p][q] != arr[p - 1][q]) {

is = true;

break;

}

}

if (is)

max -= 1;

System.out.println(max \* max);

}

private static void baosou(int a, int b, int c, int d) {

if (a >= x || b >= y) {

if (max < d) {

p = a;

q = b;

max = d;

}

return;

}

if (arr[a + 1][b] == c && arr[a][b + 1] == c && arr[a + 1][b + 1] == c) {

baosou(a + 1, b + 1, c, d + 1);

} else {

if (max < d) {

max = d;

p = a;

q = b;

}

return;

}

}

}