## POJ-2155 Matrix 解题报告

#### 题目

**描述**

Given an N\*N matrix A, whose elements are either 0 or 1. A[i, j] means the number in the i-th row and j-th column. Initially we have A[i, j] = 0 (1 <= i, j <= N).  
  
We can change the matrix in the following way. Given a rectangle whose upper-left corner is (x1, y1) and lower-right corner is (x2, y2), we change all the elements in the rectangle by using "not" operation (if it is a '0' then change it into '1' otherwise change it into '0'). To maintain the information of the matrix, you are asked to write a program to receive and execute two kinds of instructions.  
  
1. C x1 y1 x2 y2 (1 <= x1 <= x2 <= n, 1 <= y1 <= y2 <= n) changes the matrix by using the rectangle whose upper-left corner is (x1, y1) and lower-right corner is (x2, y2).  
2. Q x y (1 <= x, y <= n) querys A[x, y].

**输入**

The first line of the input is an integer X (X <= 10) representing the number of test cases. The following X blocks each represents a test case.  
  
The first line of each block contains two numbers N and T (2 <= N <= 1000, 1 <= T <= 50000) representing the size of the matrix and the number of the instructions. The following T lines each represents an instruction having the format "Q x y" or "C x1 y1 x2 y2", which has been described above.

**输出**

For each querying output one line, which has an integer representing A[x, y].  
  
There is a blank line between every two continuous test cases.

**样例输入**

1

2 10

C 2 1 2 2

Q 2 2

C 2 1 2 1

Q 1 1

C 1 1 2 1

C 1 2 1 2

C 1 1 2 2

Q 1 1

C 1 1 2 1

Q 2 1

**样例输出**

1

0

0

1

**来源**

POJ Monthly,Lou Tiancheng

#### 题意梗概

对于一个由0、1组成的矩阵，初始值全部为0，对其进行进行区间修改（0变1,1变0）及单点查询。

#### 思路

由于要对0-1矩阵进行维护，不放另设数组a,每次修改操作对a数组修改区间内的元素值**+1**，并在查询操作时对a数组内的值模2即为答案。

现在需要求如何对a数组进行区间修改及单点查询

**1.线段树**

**略......**

**2.二维树状数组+差分**

a.单点查询

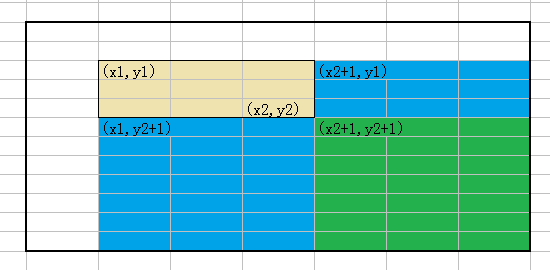
不妨设数组b为差分数组，

定义

又因为a[1][1]=0（初始为0）

对a[n][m]的询问 <==> （二维区间和）

b.区间修改



如图，不妨设我们修改的区间是浅黄色区域。

对于b数组，当修改b[x0][y0]的值时，a[i][j] ( x0<=i<=n , y0<=j<=n )的值均会被修改

得到公式：

b[x1][y1]+=1 , b[x1][y2+1]-=1 , b[x2+1][y1]-=1 , b[x2+1][y2+1]+=1;

#### 总结

用二维树状数组维护b数组，当有C操作时按b点对树状数组上的点进行四个点的单点修改，当有Q操作时按a操作求到(x,y)的区间和后模2即可

#### 参考代码

#include<bits/stdc++.h>

#define maxn 1007

using namespace std;

int n,t[maxn][maxn],x,y,xp,yp,xq,yq,T,m;

int Lowbit(int x)

{

return x&(-x);

}

void Edit(int x,int y,int val)

{

for (int i=x;i<=n;i+=Lowbit(i))

for (int j=y;j<=n;j+=Lowbit(j))

t[i][j]+=val;

}

int Query(int x,int y)

{

int ans=0;

for (int i=x;i>0;i-=Lowbit(i))

for (int j=y;j>0;j-=Lowbit(j))

ans+=t[i][j];

return ans;

}

int main()

{

scanf("%d",&T);

while (T)

{

T--;

scanf("%d%d",&n,&m);

memset(t,0,sizeof(t)); //每组数据都需先初始化

for (int i=1;i<=m;i++)

{

char c=getchar();

while (c>'Z'||c<'A') c=getchar();

if (c=='C')

{

scanf("%d%d%d%d",&xp,&yp,&xq,&yq);

Edit(xp,yp,1); Edit(xq+1,yq+1,1); Edit(xp,yq+1,-1); Edit(xq+1,yp,-1); //单点修改

}

else

{

scanf("%d%d",&x,&y);

printf("%d\n",Query(x,y)%2); //区间查询

}

}

if (T) printf("\n"); //每组数据间空行

}

return 0;

}

#### 格式提醒

题目含多组数据，对于每两组数据之间需有一行空行