## 二维树状数组推导（区间修改区间查询）

对于一个差分矩阵数组$d[n][m]$，原数组$a[n][m]$，求 $(x,y)$ 点的对应矩阵的和， 公式为

$$\sum_{i}^{x}\sum_{j}^{y}\sum_{u}^{i}\sum_{v}^{j}d[u][v]$$

其中

$$\begin{aligned}
&a[1][1]=d[1][1]\\
&a[1][2]=d[1][1]+d[1][2]\\
&a[1][3]=d[1][1]+d[1][2]+d[1][3]\\
&a[1][y]=d[1][1]+d[1][2]+d[1][3]+\ldots+d[1][y]\\
&a[2][1]=d[1][1]+d[2][1]\\
&a[2][2]=d[1][1]+d[1][2]+d[2][1]+d[2][2]\\
&a[2][3]=d[1][1]+d[1][2]+d[1][3]+d[2][1]+d[2][2]+d[2][3]\\
&a[2][y]=d[1][1]+d[1][2]+d[1][3]+\ldots+d[1][y]+d[2][1]+d[2][2]+d[2][3]+\ldots+d[2][y]\\
&\qquad\ldots\ldots
\end{aligned}$$

经过观察可以得到，在求 $(x,y)$ 点的对应矩阵的和中， $d[1][1]$出现了 $x*y$ 次， $d[1][2]$ 出现了 $x*(y-1)$次，$d[1][3]$出现了$x*(y-2)$次， $d[1][y]$出现了 $x*(y-y+1)$次， 类比得$d[u][v]$出现了$(x-u+1)*(y-v+1)$次（也可以用排列组合来理解）， 因此上述公式可以化简为

$$\sum_{i}^{x}\sum_{j}^{y}d[i][j]*(x-i+1)*(y-j+1)$$

展开得

$$\begin{aligned}
&\sum_{i}^{x}\sum_{j}^{y}a[i][j]\\
&=((x+1)*(y+1)-i*(y+1)-j*(x+1)+i*j)\sum_{i}^{x}\sum_{j}^{y}d[i][j]\\
&=(x+1)(y+1)\sum_{i}^{x}\sum_{j}^{y}d[i][j]-(y+1)\sum_{i}^{x}\sum_{j}^{y}d[i][j]*i-(x+1)\sum_{i}^{x}\sum_{j}^{y}d[i][j]*j+\sum_{i}^{x}\sum_{j}^{y}d[i][j]*i*j
\end{aligned}$$

以上便是求$(x,y)$点对应矩阵的和的公式了；

因此，若要求 $(x,y)$ 点的对应矩阵的和，我们维护四个树状数组即可， 分别是$d[i][j],d[i][j]*i,d[i][j]*j,d[i][j]*i*j$;

以下用一道模板题P4514上帝造题的七分钟来展示代码

```
#include<bits/stdc++.h>

using namespace std;
using LL = long long ;

const int N = 2050 , M = 2050 ;

int n , m ;
int tr1[N][M] ;
LL tr2[N][M] , tr3[N][M] , tr4[N][M];

int lowbit(int x)
{
```

```
    return x & -x;
}

void add(int x ,int y ,int c)
{
   for(int i = x ; i <= n ; i += lowbit(i))
       for(int j = y ; j <= m ; j += lowbit(j))
       {
            tr1[i][j] += c;
            tr2[i][j] += 1ll * x * c;
            tr3[i][j] += 1ll * y * c;
            tr4[i][j] += 1ll * x * y * c;
        }
}
LL sum(int x , int y)
{
    LL res = 0;
    for(int i = x ; i ; i -= lowbit(i))
        for(int j = y ; j ; j -= lowbit(j))
         {
             res += 1ll * (x + 1) * (y + 1) * tr1[i][j];
             res -= 1ll * (y + 1) * tr2[i][j];
             res -= 1ll * (x + 1) * tr3[i][j];
             res += 1ll * tr4[i][j];
         }
    return res;
}

void modify(int x1 , int y1 , int x2 ,int y2 , int d)
{
    add(x1 , y1 , d);
    add(x1 , y2 + 1 , -d);
    add(x2 + 1, y1 , -d);
    add(x2 + 1 , y2 + 1 , d);
}


LL query(int x1 , int y1 , int x2 , int y2)
{
    return sum(x2 , y2) - sum(x1 - 1 , y2) - sum(x2 , y1 - 1) + sum(x1 - 1 , y1 - 1);
}

int main()
{
    ios::sync_with_stdio(false);
    cin.tie(nullptr);

    char x;
    cin >> x >> n >> m;

    string op;
    while(cin >> op)
    {
        int x1 , y1 , x2 , y2;
        cin >> x1 >> y1 >> x2 >> y2;
        if(op == "L")
        {
            int d;cin >> d;
            modify(x1 , y1 , x2 , y2 , d);
        }
        else
```

```
        {
            cout << query(x1 , y1 , x2 , y2) << "\n";
        }
    }

    return 0 ;
}
```