**[树状数组学习心得](http://www.cppblog.com/linyangfei/archive/2008/09/24/62688.html)**

POJ 3321 Apple Tree

<http://acm.pku.edu.cn/JudgeOnline/problem?id=3321>

POJ 2481 Cows

<http://acm.pku.edu.cn/JudgeOnline/problem?id=2481>

POJ 2155 Matrix

<http://acm.pku.edu.cn/JudgeOnline/problem?id=2155>

       今天在POJ淘了这几道题目，学习了一下树状数组的用法，跟大家分享一下心得。可以把树状数组看成一种数据结构，对于一个数组，如果有多次操作，每次的操作有两种：1、修改数组中某一元素的值，2、求和，求数组元素a[1]+a[2]+…a[num]的和。这是树状数组最基本的应用了，用树状数组可以实现O(log(n))的修改以及O(log(n))的求和。当然用线段树完全可以胜任这些计算，但是线段树写起来代码比较长，并且线段树要占用2\*n大小的空间。下面先给出树状数组的三个基本操作的函数：

int lowbit(int k)

{

       return k&(-k);

}

//lowbit函数是计算k的二进制位最低位不为0的数字的权值。

void Modify(int num, int v)

{

       while(num <= n)

       {

              c[num]+=v;

              num+=lowbit(num);

       }

}

//Modify函数是往数组c中修改值，更新整个数组的值，实现了操作1；

int Sum(int num)

{

       int ans=0;

       while(num > 0)

       {

              ans+=c[num];

              num-=lowbit(num);

       }

       return ans;

}

//Sum函数返回数组元素a[1]+a[2]+…a[num]的和，实现操作2；

       树状数组的巧妙之处在于对于数组下标的二进制的操作，假设a[1...N]为原数组,定义c[1...N]为对应的树状数组:

c[i] = a[i - 2^k + 1] + a[i - 2^k + 2] + ... + a[i]

其中k为i的二进制表示末尾0的个数,所以2^k即为i的二进制表示的最后一个1的权值.

可以把树状数组看作是把数组分成了若干个2^k大小的空间。对于一个下标i，c[i]的值是由i/(lowbit(i))个数组元素的值所组成的，每次步进的单位是k=lowbit(i)，这个有点像二分归并的思想！这样就可以实现O(log(n))的修改和查询。

       下面是树状数组的具体应用：

       3321 Apple Tree 一棵树上长了苹果，每一个树枝节点上有长苹果和不长苹果两种状态，两种操作，一种操作能够改变树枝上苹果的状态，另一种操作询问某一树枝节点一下的所有的苹果有多少。具体做法是做一次dfs，记下每个节点的开始时间low[i]和结束时间high[i]，那么对于i节点的所有子孙的开始时间和结束时间都应位于low[i]和high[i]之间，另外用一个数组c[i]记录附加在节点i上的苹果的个数，然后用树状数组统计low[i]到high[i]之间的附加苹果总数。这里用树状数组统计区间可以用Sum(high[i])-Sum(low[i]-1)来计算。

http://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gif  
 1http://www.cppblog.com/Images/OutliningIndicators/None.gif#include <stdio.h>  
 2http://www.cppblog.com/Images/OutliningIndicators/None.gif#include <string.h>  
 3http://www.cppblog.com/Images/OutliningIndicators/None.gif#include <vector>  
 4http://www.cppblog.com/Images/OutliningIndicators/None.gifusing namespace std;  
 5http://www.cppblog.com/Images/OutliningIndicators/None.gif  
 6http://www.cppblog.com/Images/OutliningIndicators/None.gif//vector<int> g[100005];  
 7http://www.cppblog.com/Images/OutliningIndicators/None.gifstruct Node  
 8http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
 9http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int v;  
10http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    struct Node \*next;  
11http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}g[100005];  
12http://www.cppblog.com/Images/OutliningIndicators/None.gifint n,m,cnt,low[100005],high[100005],c[100005],flag[100005];  
13http://www.cppblog.com/Images/OutliningIndicators/None.gifbool mark[100005];  
14http://www.cppblog.com/Images/OutliningIndicators/None.gif  
15http://www.cppblog.com/Images/OutliningIndicators/None.gifvoid dfs(int v)  
16http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
17http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    struct Node \*p=g[v].next;  
18http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    mark[v]=true;  
19http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    cnt++;  
20http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    low[v]=cnt;  
21http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    while(p)  
22http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
23http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(!mark[p->v])  
24http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            dfs(p->v);  
25http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        p=p->next;  
26http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
27http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    high[v]=cnt;  
28http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
29http://www.cppblog.com/Images/OutliningIndicators/None.gifint lowbit(int k)  
30http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
31http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return k&(-k);  
32http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
33http://www.cppblog.com/Images/OutliningIndicators/None.gifvoid Modify(int num, int v)  
34http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
35http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    while(num <= n)  
36http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
37http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        c[num]+=v;  
38http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        num+=lowbit(num);  
39http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
40http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
41http://www.cppblog.com/Images/OutliningIndicators/None.gifint Sum(int num)  
42http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
43http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int ans=0;  
44http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    while(num > 0)  
45http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
46http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        ans+=c[num];  
47http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        num-=lowbit(num);  
48http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
49http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return ans;  
50http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
51http://www.cppblog.com/Images/OutliningIndicators/None.gif  
52http://www.cppblog.com/Images/OutliningIndicators/None.gifint main()  
53http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
54http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i,j,a,b,ans;  
55http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    char temp[10];  
56http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    struct Node \*p;  
57http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    //freopen("in.txt","r",stdin);  
58http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    scanf("%d",&n);  
59http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    memset(g,0,sizeof(g));  
60http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    for(i=1; i<n; i++)  
61http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
62http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        scanf("%d%d",&a,&b);  
63http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        p=new Node;  
64http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        p->next=g[a].next;  
65http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        p->v=b;  
66http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        g[a].next=p;  
67http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        p=new Node;  
68http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        p->next=g[b].next;  
69http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        p->v=a;  
70http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        g[b].next=p;  
71http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
72http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    memset(mark,false,sizeof(mark));  
73http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    memset(c,0,sizeof(c));  
74http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    for(i=1; i<=n; i++)  
75http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        flag[i]=1;  
76http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    cnt=0;  
77http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    dfs(1);  
78http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    scanf("%d",&m);  
79http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    while(m--)  
80http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
81http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        scanf("%s",temp);  
82http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(temp[0] == 'Q')  
83http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
84http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            scanf("%d",&a);  
85http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            ans=high[a]-low[a]+1+Sum(high[a])-Sum(low[a]-1);  
86http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            printf("%d\n",ans);  
87http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
88http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        else  
89http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
90http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            scanf("%d",&a);  
91http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            if(flag[a]) Modify(low[a],-1);  
92http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            else Modify(low[a],1);  
93http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            flag[a]^=1;  
94http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
95http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
96http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return 0;  
97http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
98http://www.cppblog.com/Images/OutliningIndicators/None.gif  
99http://www.cppblog.com/Images/OutliningIndicators/None.gif

       2481 Cows 给n个区间[Si,Ei]，区间[Sj,Ej]< [Si,Ei] 有 Si <= Sj and Ej <= Ei and Ei - Si > Ej – Sj。按y坐标从小到达，x坐标从大到小的顺序排序，然后从后往前扫描，记录i之前所有的j区间Sj<Si的个数，这个用树状数组实现。扫描一遍可得出结果。

http://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gif  
 1http://www.cppblog.com/Images/OutliningIndicators/None.gif#include <stdio.h>  
 2http://www.cppblog.com/Images/OutliningIndicators/None.gif#include <string.h>  
 3http://www.cppblog.com/Images/OutliningIndicators/None.gif#include <algorithm>  
 4http://www.cppblog.com/Images/OutliningIndicators/None.gifusing namespace std;  
 5http://www.cppblog.com/Images/OutliningIndicators/None.gif  
 6http://www.cppblog.com/Images/OutliningIndicators/None.gifstruct P  
 7http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
 8http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int x,y,id;  
 9http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}p[100005];  
10http://www.cppblog.com/Images/OutliningIndicators/None.gifint n,a[100005],max\_n,b[100005];  
11http://www.cppblog.com/Images/OutliningIndicators/None.gif  
12http://www.cppblog.com/Images/OutliningIndicators/None.gifint lowbit(int k)  
13http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
14http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return k&(-k);  
15http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
16http://www.cppblog.com/Images/OutliningIndicators/None.gifvoid Modify(int num, int v)  
17http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
18http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    while(num <= max\_n)  
19http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
20http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        a[num]+=v;  
21http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        num+=lowbit(num);  
22http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
23http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
24http://www.cppblog.com/Images/OutliningIndicators/None.gifint Sum(int num)  
25http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
26http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int ans=0;  
27http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    if(num <= 0) return 0;  
28http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    while(num)  
29http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
30http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        ans+=a[num];  
31http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        num-=lowbit(num);  
32http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
33http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return ans;  
34http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
35http://www.cppblog.com/Images/OutliningIndicators/None.gifbool operator <(const P a, const P b)  
36http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
37http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    if(a.y == b.y) return a.x > b.x;  
38http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return a.y < b.y;  
39http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
40http://www.cppblog.com/Images/OutliningIndicators/None.gif  
41http://www.cppblog.com/Images/OutliningIndicators/None.gifint main()  
42http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
43http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i;  
44http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    //freopen("in.txt","r",stdin);  
45http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    while(scanf("%d",&n), n)  
46http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
47http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        max\_n=0;  
48http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        for(i=0; i<n; i++)  
49http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
50http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            scanf("%d%d",&p[i].x,&p[i].y);  
51http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            p[i].id=i;  
52http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            p[i].x++;  
53http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            p[i].y++;  
54http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            if(p[i].y > max\_n) max\_n=p[i].y;  
55http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
56http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        sort(p,p+n);  
57http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        memset(a,0,sizeof(a));  
58http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        for(i=n-1; i>=0; i--)  
59http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
60http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            if(i != n-1 && p[i].y == p[i+1].y && p[i].x == p[i+1].x)  
61http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                b[p[i].id]=b[p[i+1].id];  
62http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            else  
63http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                b[p[i].id]=Sum(p[i].x);  
64http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            Modify(p[i].x,1);  
65http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
66http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        for(i=0; i<n; i++)  
67http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
68http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            if(i) printf(" ");  
69http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            printf("%d",b[i]);  
70http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
71http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        printf("\n");  
72http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
73http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return 0;  
74http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
75http://www.cppblog.com/Images/OutliningIndicators/None.gif  
76http://www.cppblog.com/Images/OutliningIndicators/None.gif

       2155 Matrix 有n\*n的0，1矩阵，两种操作，1、翻转矩形（x1,y1）(x2,y2)的值，2、输出位置为(x,y)矩阵处的值。先考虑一维的情况，设A<B，那么要翻转[A,B]之间的值，可以分解为两步操作，先翻转[1,A-1]，然后再翻转[1,B]，其中翻转的次数就可以用树状数组来计算。然后再将次操作扩展到二维的情形，只需将x方向与y方向套成一个二重循环即可。从这里我们也可以看到树状数组处理类似问题时比线段树的优越性。从代码的长度，空间消耗上面，树状数组都有明显的优势。

http://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gif  
 1http://www.cppblog.com/Images/OutliningIndicators/None.gif#include <stdio.h>  
 2http://www.cppblog.com/Images/OutliningIndicators/None.gif#include <string.h>  
 3http://www.cppblog.com/Images/OutliningIndicators/None.gif  
 4http://www.cppblog.com/Images/OutliningIndicators/None.gifint a[1005][1005],n,m;  
 5http://www.cppblog.com/Images/OutliningIndicators/None.gif  
 6http://www.cppblog.com/Images/OutliningIndicators/None.gifint lowbit(int k)  
 7http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
 8http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return k&(-k);  
 9http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
10http://www.cppblog.com/Images/OutliningIndicators/None.gifvoid Modify(int x1, int y1, int x2, int y2)  
11http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
12http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i,j;  
13http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    for(i=x1-1; i>0; i-=lowbit(i))  
14http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
15http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        for(j=y1-1; j>0; j-=lowbit(j))  
16http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
17http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            a[i][j]^=1;  
18http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
19http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        for(j=y2; j>0; j-=lowbit(j))  
20http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
21http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            a[i][j]^=1;  
22http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
23http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
24http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    for(i=x2; i>0; i-=lowbit(i))  
25http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
26http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        for(j=y1-1; j>0; j-=lowbit(j))  
27http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
28http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            a[i][j]^=1;  
29http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
30http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        for(j=y2; j>0; j-=lowbit(j))  
31http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
32http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            a[i][j]^=1;  
33http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
34http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
35http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
36http://www.cppblog.com/Images/OutliningIndicators/None.gifint Sum(int x, int y)  
37http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
38http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int ans=0,i,j;  
39http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    for(i=x; i<=n; i+=lowbit(i))  
40http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
41http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        for(j=y; j<=n; j+=lowbit(j))  
42http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            ans^=a[i][j];  
43http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
44http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return ans;  
45http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
46http://www.cppblog.com/Images/OutliningIndicators/None.gif  
47http://www.cppblog.com/Images/OutliningIndicators/None.gifint main()  
48http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifhttp://www.cppblog.com/Images/dot.gif{  
49http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i,j,x1,x2,y1,y2,cases,ic=0;  
50http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    char temp[10];  
51http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    //freopen("in.txt","r",stdin);  
52http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    scanf("%d",&cases);  
53http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    while(cases--)  
54http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    http://www.cppblog.com/Images/dot.gif{  
55http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if(ic++) printf("\n");  
56http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        scanf("%d%d",&n,&m);  
57http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        memset(a,0,sizeof(a));  
58http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        while(m--)  
59http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        http://www.cppblog.com/Images/dot.gif{  
60http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            scanf("%s",temp);  
61http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            if(temp[0] == 'C')  
62http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cppblog.com/Images/dot.gif{  
63http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                scanf("%d%d%d%d",&x1,&y1,&x2,&y2);  
64http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                Modify(x1,y1,x2,y2);  
65http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
66http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            else  
67http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif            http://www.cppblog.com/Images/dot.gif{  
68http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                scanf("%d%d",&x1,&y1);  
69http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                printf("%d\n",Sum(x1,y1));  
70http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
71http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
72http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
73http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return 0;  
74http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
75http://www.cppblog.com/Images/OutliningIndicators/None.gif  
76http://www.cppblog.com/Images/OutliningIndicators/None.gif