线段树

by : SHM&CZY

目录

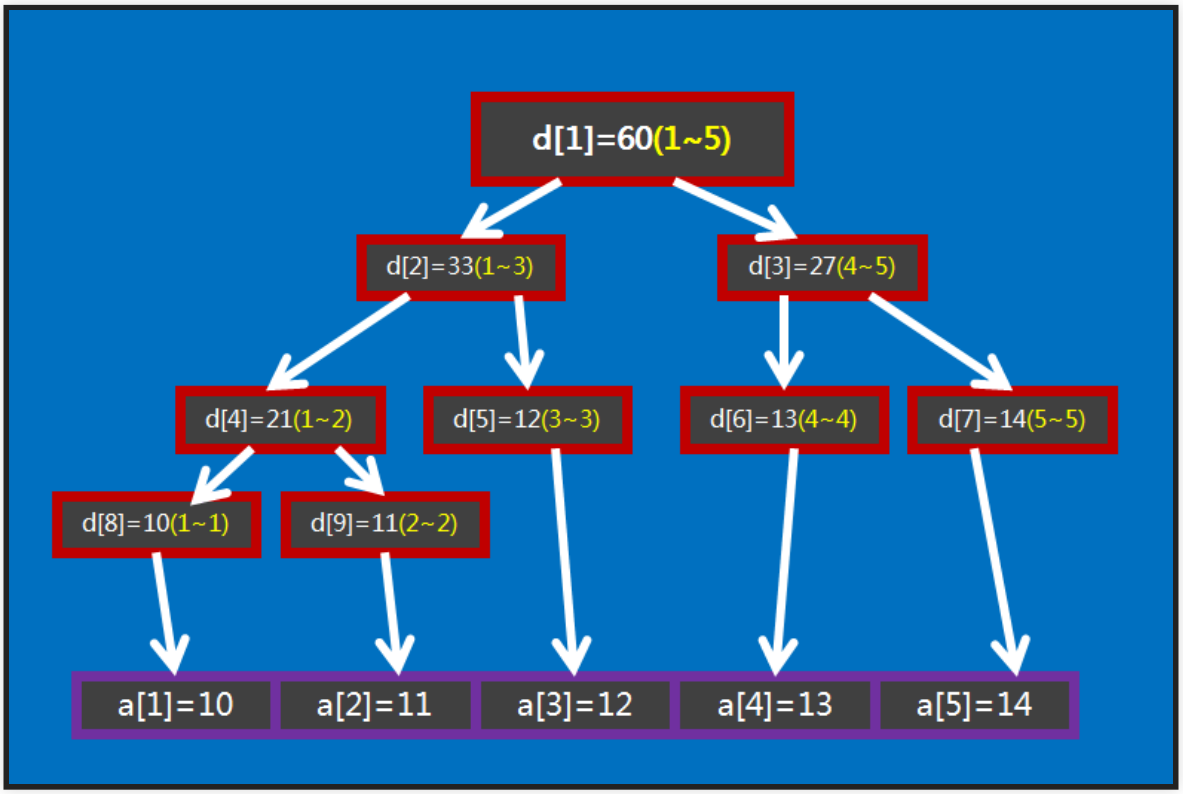
线段树简介

- 线段树是信息学竞赛中常用的维护**区间信息**的数据结构。
- 线段树可以在$\mathcal{O}(\log \mathcal{N})$的时间复杂度内实现单点修改、查询，区间修改、查询（求和，求min, max）等操作。

线段树简介

- 线段树维护的信息需要满足可加性，即以可接受的复杂度进行信息的合并与修改，包括在使用懒标记时标记也要满足可加性（如**取模**操作就不满足可加性，先对a取模再对b取模显然就不能合并在一起操作）。

普通线段树



普通线段树的形态

普通线段树

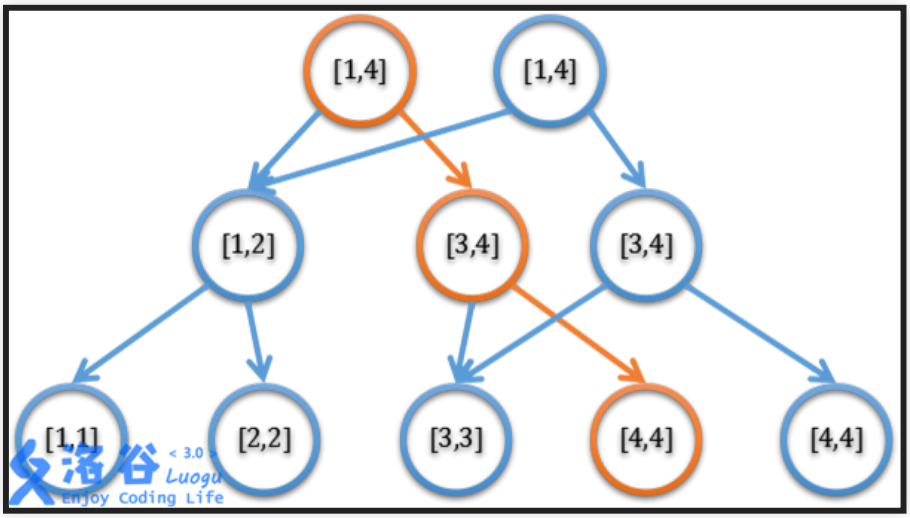
线段树的空间复杂度：$\mathcal{O}(4n)$

主席树

主席树

简介

主席树原名**可持久化线段树**。可以给普通的线段树增加一些历史节点来维护历史数据，用较优的复杂度访问历史数据。



主席树

与普通线段树的区别

需要动态开点，事权值线段树（基本上）

主席树

优化空间复杂度

每一个历史版本都建立一棵完整的线段树空间是不够的。如果当前版本只修改了左子树，那么当前版本的右子树可以使用它前驱的右子树。

主席树

例题

洛谷$P3834$维护区间第K大

给定n个整数构成的序列a，将对于指定的闭区间$[l,r]$查询其区间内的第k小值，有m次询问。

$n,m \leq 2*10^5$

主席树

例题

洛谷$P3834$维护区间第K大

给定n个整数构成的序列a，将对于指定的闭区间$[l,r]$查询其区间内的第k小值，有m次询问。

$n,m \leq 2*10^5$

主席树

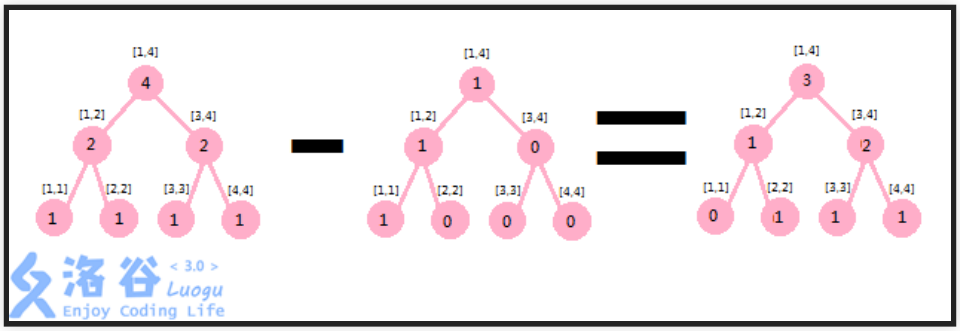
例题

洛谷$P3834$维护区间第K大

给定n个整数构成的序列a，将对于指定的闭区间$[l,r]$查询其区间内的第k小值，有m次询问。

$$n,m \leq 2*10^5$$

解析：



利用前缀和与二分的思想，通过区间查询来检验是否为第k大。

主席树

例题

[$CQOI2015$]任务查询系统

现在有m个有权值的任务，将每个任务放置在一段区间内，单点查询某个点上权值前k大的权值总和。

$m \leq 100000$

主席树

例题

[$CQOI2015$]任务查询系统

现在有m个有权值的任务，将每个任务放置在一段区间内，单点查询某个点上权值前k大的权值总和。

$m \leq 100000$

解析：

利用差分的思想，把区间修改转换为单点修改区间查询

线段树合并

线段树合并

简介

之前说过，线段树维护的信息具有可加性，也就是说对应节点代表的区间是一样的，只是保存的信息不同。于是我们就可以把两颗线段树的信息进行合并。

线段树合并

例题

给出一棵树，每个点有一个权值，求在一个点的子树中，有多少个点的权值比它大。

$n \leq 10^5$

线段树合并

例题

给出一棵树，每个点有一个权值，求在一个点的子树中，有多少个点的权值比它大。

$n \leq 10^5$

解析：

每个点建立一棵权值线段树，从根节点dfs，然后从下往上不断线段树合并即可。

线段树合并

例题

给出一棵有n个节点树，给定两个数p, k，p表示询问p号节点，询问的内容是：有多少个三元组(a, p, c)满足c的祖先是a, p节点，并且a, p两个点间的距离不超过k。共有q个询问。

$n \leq 3000000, q \leq 3000000, 1 \leq p, k \leq n$

语文不行，题意有问题看原题面吧

线段树合并

例题

解析

这道题就是给定点a，求一个与a距离不超过某一个值的点b，并且使另一个点c是a,b的后代。

如果a,b没有祖先后代关系则没有答案。

如果b是a的祖先容易知道$ans=\min(k,dep[a]-1)*(size[a]-1)$,$size$表示子树大小

如果a是b的祖先，则方案数是深度在$dep[a]+1$到$dep[a]+k$中每一个b的$size[b]-1$

线段树合并

例题

解析

对于第一种操作直接计算，第二种可以考虑用线段树合并，下标为节点深度，维护的值是$size-1$。第二种情况即为4求$(dep[a]+1,dep[a]+k)$的区间和

线段树合并

例题

解析

但是值得注意的是，平时我们的线段树合并都是把两个节点的信息合并到一个已经存在的节点上，但是这道题为了不改变儿子线段树的值，所以在合并的时候要新建节点。

线段树合并

例题

解析

显然对于交换子树的操作，第一种和第二种是没有意义的，因为先序遍历的顺序是根→左子树→右子树。所以只考虑跨越子树的情况。

线段树合并

例题

解析

对于一个点，设它的左子树是p，右子树是q，则有

1.如果不交换子树：$ans1 = [p.lson].size * [q.rson].size$

2.如果交换子树：$ans2 = [p.rson].size * [q.lson].size$

最终的答案加上$min(ans1, ans2)$

时间线段树

时间线段树

简介

又称线段树分治，用来处理离线问题，一般而言，要用到它的题目里的操作都会有一个有效区间。

使用方法大家也都知道，就是把操作离线出来放到一棵时间线段树上，作为区间修改，询问答案就是在线段树上单点查询。在使用时间线段树的时候，我们常常要用到一些支持撤销的数据结构。

时间线段树

P5787 二分图 /【模板】线段树分治

给一个图，有删边加边，问每一个操作后这张图是不是二分图。

时间线段树

P5787 二分图 /【模板】线段树分治

把操作用vector挂到具体区间上，可撤销并查集判二分图，到叶子节点就进行询问操作。

时间线段树

CF576E PAINTING EDGES

有一张图，上面有m条边，每一个操作是把一条边染色，若同一个颜色的边构成了非二分图，我们就说这个操作不合法。

如果这个操作不合法，那么它就不会执行，问有哪些操作合法并被执行了。

时间线段树

CF576E PAINTING EDGES

和上一题差不太多，唯一的差距就是你不知道每个操作的会不会执行，以及如果执行的话右边界在哪里。

但是如果这个操作合法的话，我们是可以知道它至少会延续到下次这条边被改颜色。所以我们只需要在询问的时候判断一下操作合不合法，然后看是拿上一个颜色还是当前操作的颜色去涂这个新的区间就好了。

简而言之，就是一边判一边涂。

时间线段树

YZOJ4311 向量

你要维护一个向量集合，支持以下操作：

1.插入一个向量(x,y)

2.删除插入的第i个向量

3.查询当前集合与(x,y)点积（数量积）的最大值是多少。如果当前是空集输出0

好吧我本来是想找一道线段树分治维护斜率优化的题的。看见过这个题的人请保持沉默。

时间线段树

YZOJ4311 向量

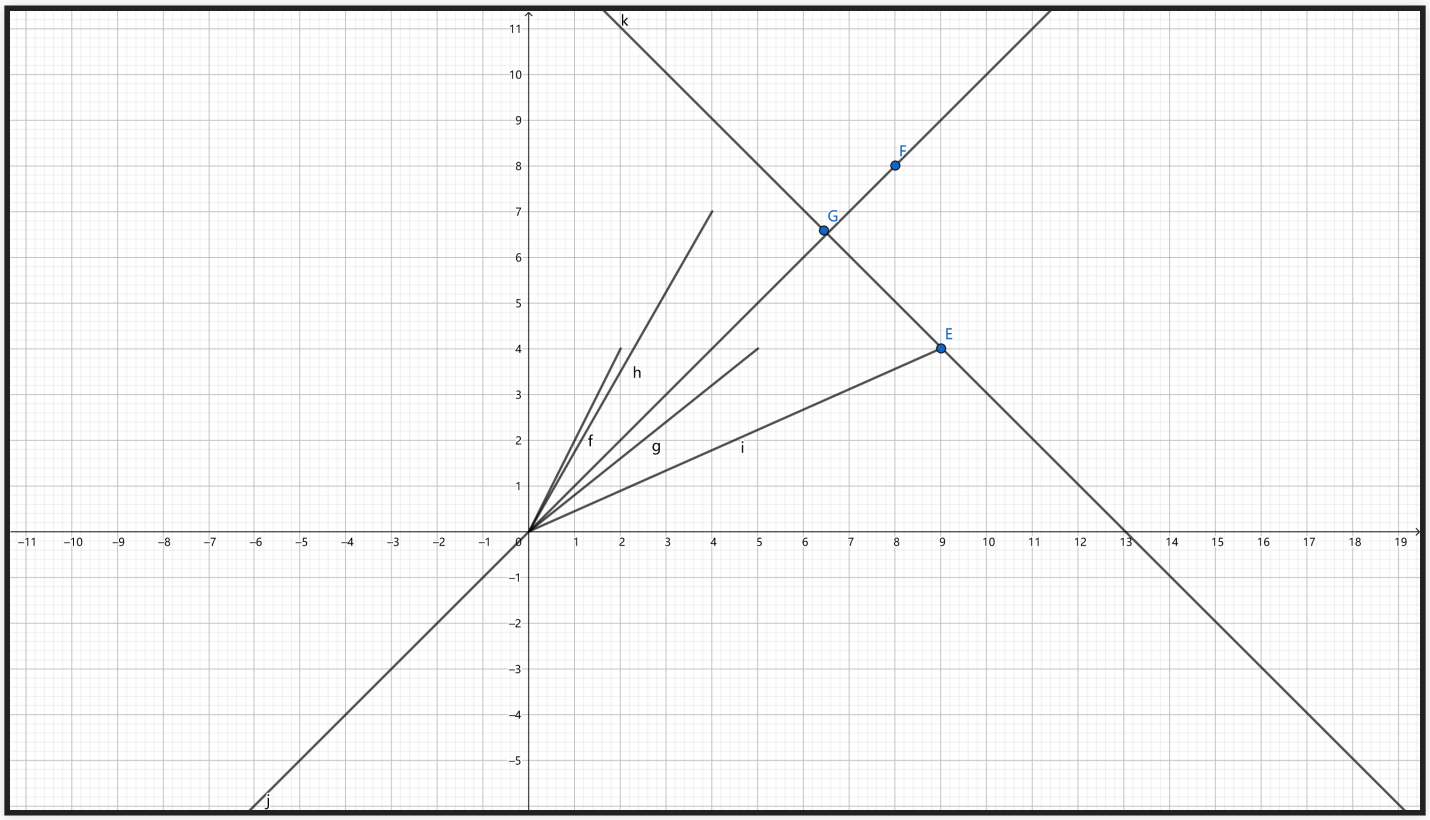
请大家自行检验自己数学必修二第一单元的知识是否牢固。

怎么样快速得知集合里最优的是哪一个向量？我们思考点积的定义：$|a||b|cos < A, B >$

一个向量的长度是固定的，那么我们只需要最大化$|a|cos < A, B >$，注意到所有向量都在第一象限

然后我们就做一条垂直于询问向量的线，不断像原点逼近，第一个接触到的集合内向量一定是最优解

然后我们就做一条垂直于询问向量的线，不断像原点逼近，第一个接触到的集合内向量一定是最优解



时间线段树

YZOJ4311 向量

于是可能作为最优解的向量一定在上凸壳上，然后我们用单调栈来维护上凸壳。

然而你在用线段树分治搞得时候插入的新向量并不一定横坐标单调递增，所以我们在插入前就排一边序，这可以保证插入进同一个区间的向量横坐标递增，所以我们单点查询的时候需要每一个遍历到的区间都询问一次，三分法或者二分法求解，复杂度$O(mlog^2n)$

其实还可以更快一点，如果我们把询问按照极角排序的话，可以发现最优解所在向量是单调递减的，然后就可以去掉一个log

扫描线

扫描线

简介

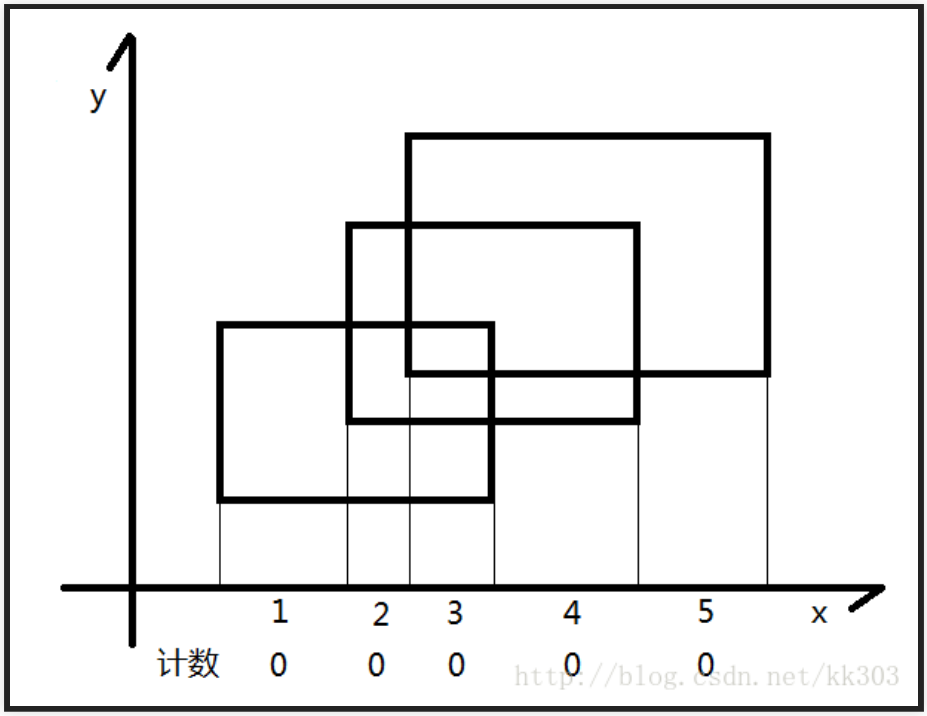
扫描线一般运用来解决图形面积问题、周长等问题。

扫描线

*ATLANTIS*问题

问题

在二维坐标系上，给出多个矩形的左下角和右上角的坐标，求所有矩形构成的图形的面积。

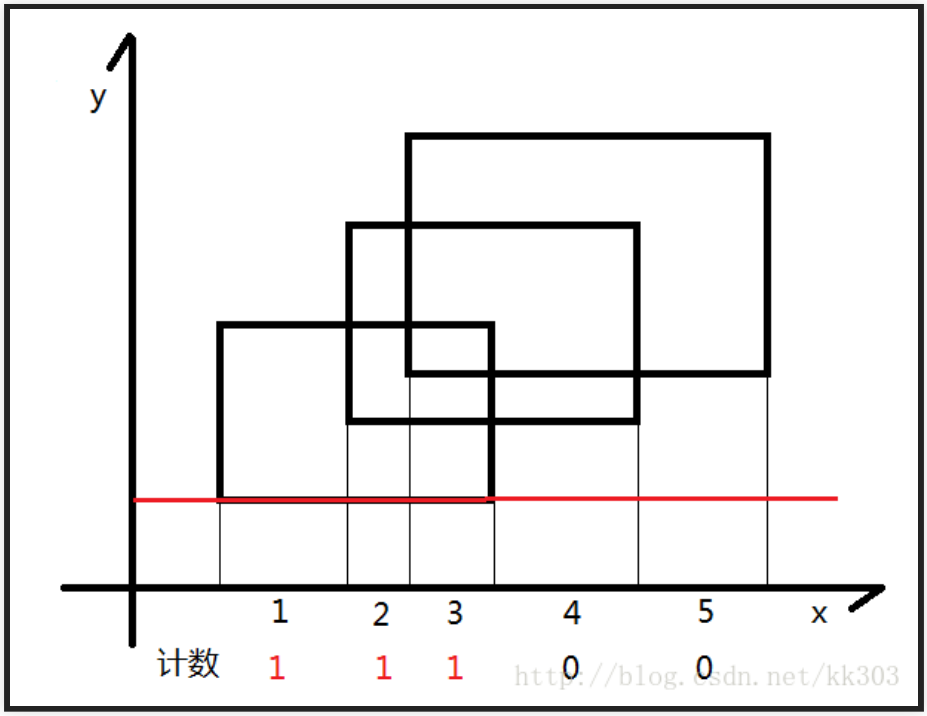


扫描线

*ATLANTIS*问题

解法

运用扫描线。假设现在有一根线从下向上扫描：

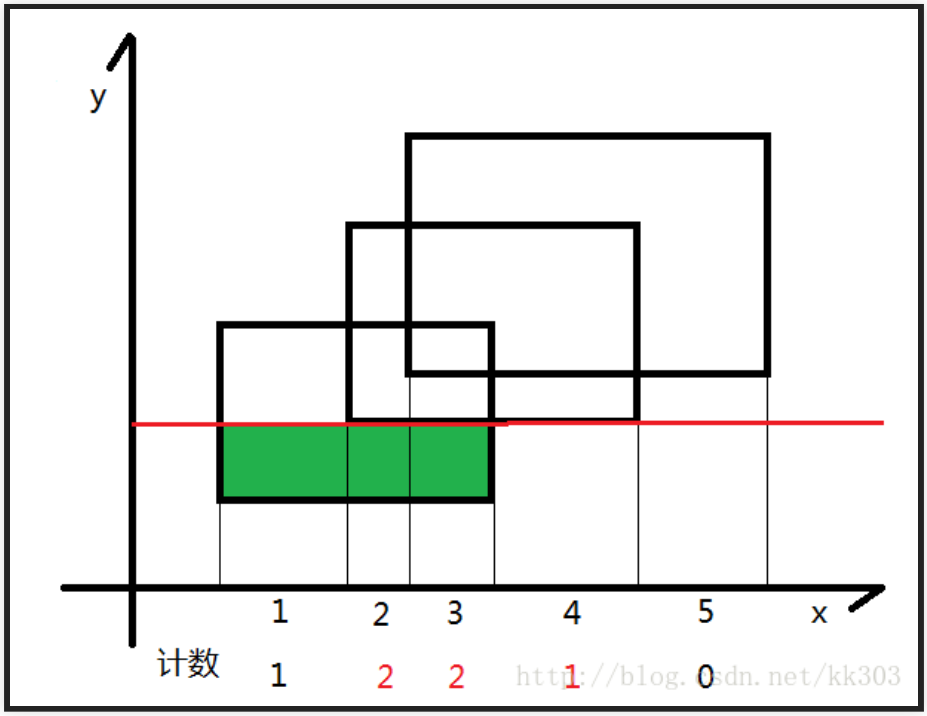


扫描线

*ATLANTIS*问题

解法

运用扫描线。假设现在有一根线从下向上扫描：

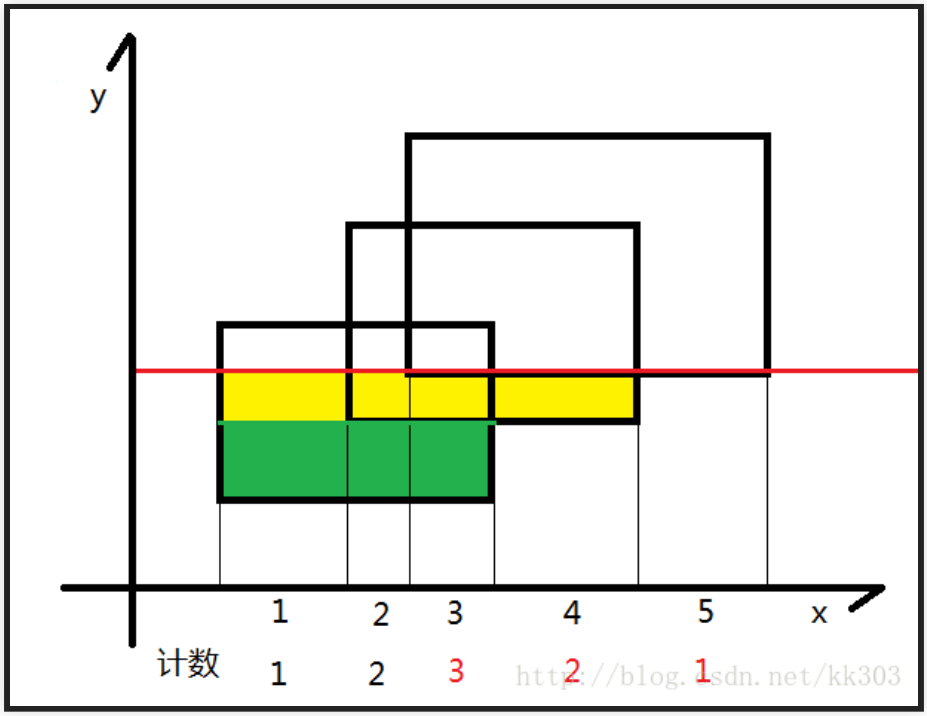


扫描线

*ATLANTIS*问题

解法

运用扫描线。假设现在有一根线从下向上扫描：

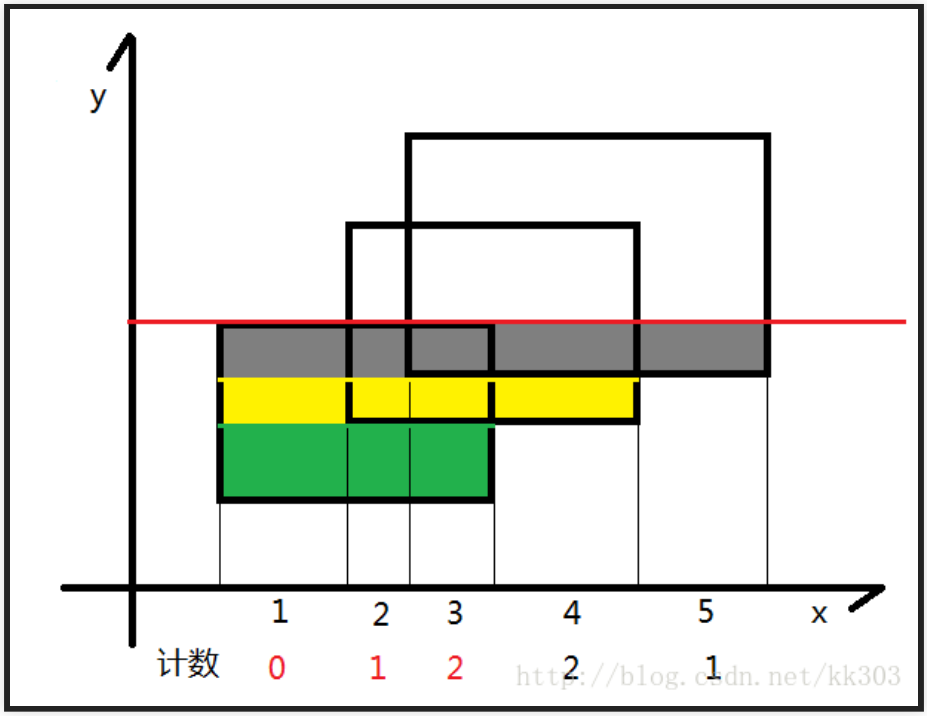


扫描线

*ATLANTIS*问题

解法

运用扫描线。假设现在有一根线从下向上扫描：

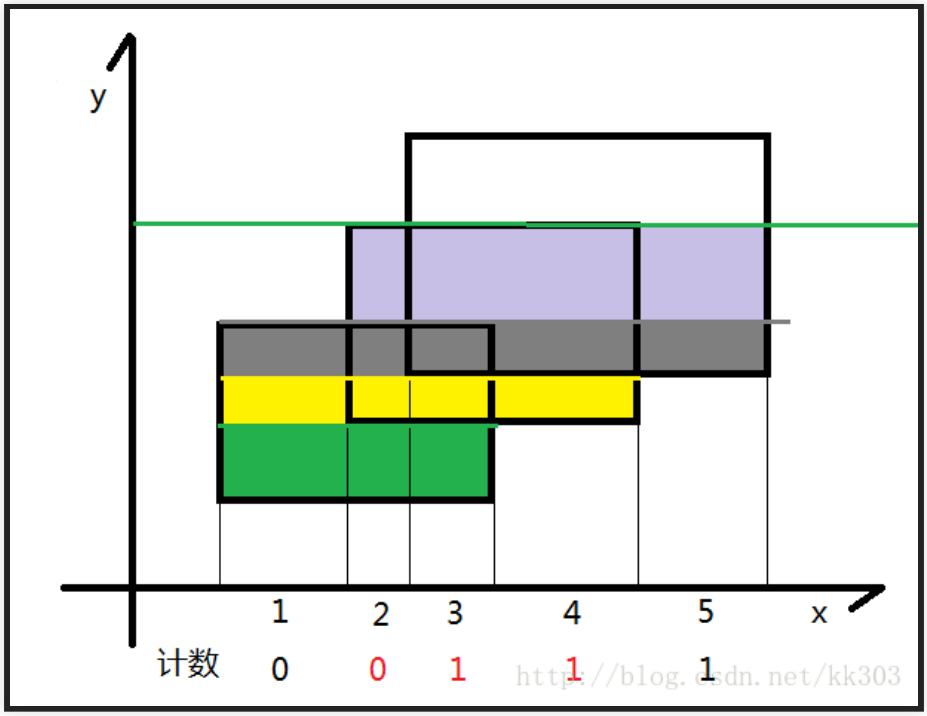


扫描线

*ATLANTIS*问题

解法

运用扫描线。假设现在有一根线从下向上扫描：

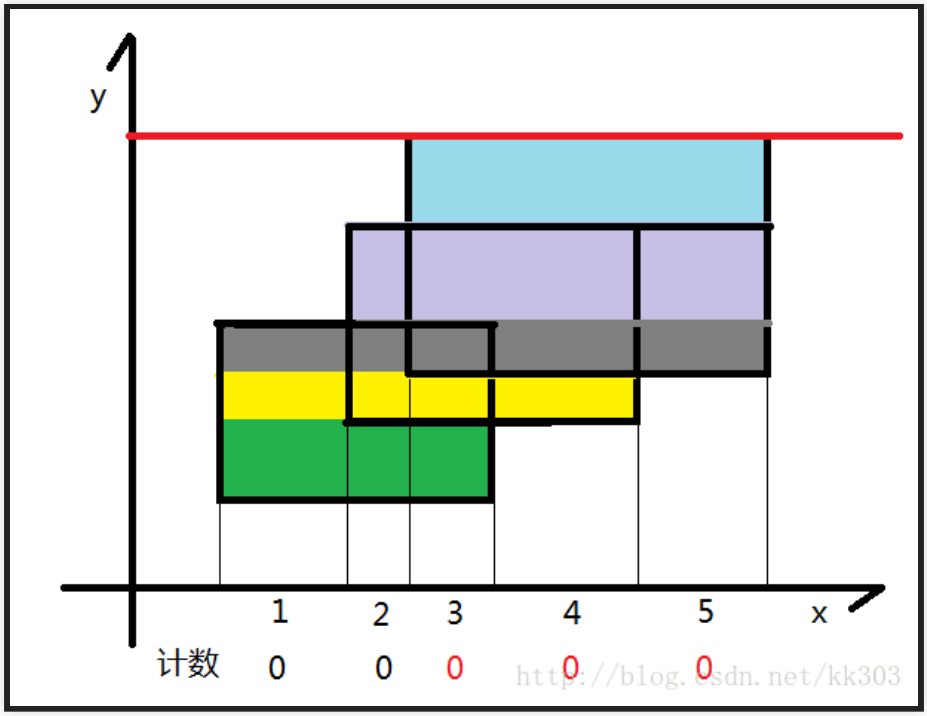


扫描线

*ATLANTIS*问题

解法

运用扫描线。假设现在有一根线从下向上扫描：



扫描线

解法

我们把整个图形分割成几个小矩形，高是扫描线走过的距离，只有长是不断变化的。于是可以利用线段树来维护各个矩形的长度。

扫描线

例题

51NOD1559 车和矩形

在棋盘（n*m）上给定一些车(k)和矩形(q)，这些车可以攻击到同行和同列，一个矩形只能被它矩形内的车给攻击到，问哪些矩形每个地方都可以被攻击到。

$1 \leq n, m \leq 100000, 1 \leq k, q \leq 200000$

扫描线

例题

51NOD1559 车和矩形

注意到只有一个矩形内只有每一行都有车或者每一列都有车才可以满足要求。

如果只考虑每一行都有的话，我们用扫描线从下向上扫，将车的纵坐标作为值单点修改到线段树里，矩形上边界作为区间询问，维护区间最小值，如果询问到区间内最小的纵坐标不在区间内，则这个区间并非每一行都有车。

纵坐标同理，颠倒横纵坐标来做就好了。

其他种类线段树

其他种类线段树

ZKW线段树

简介

zkw线段树是非递归形式的，并且**代码很短**

其他种类线段树

ZKW线段树

简介

跟线段树是**基本一毛一样**的

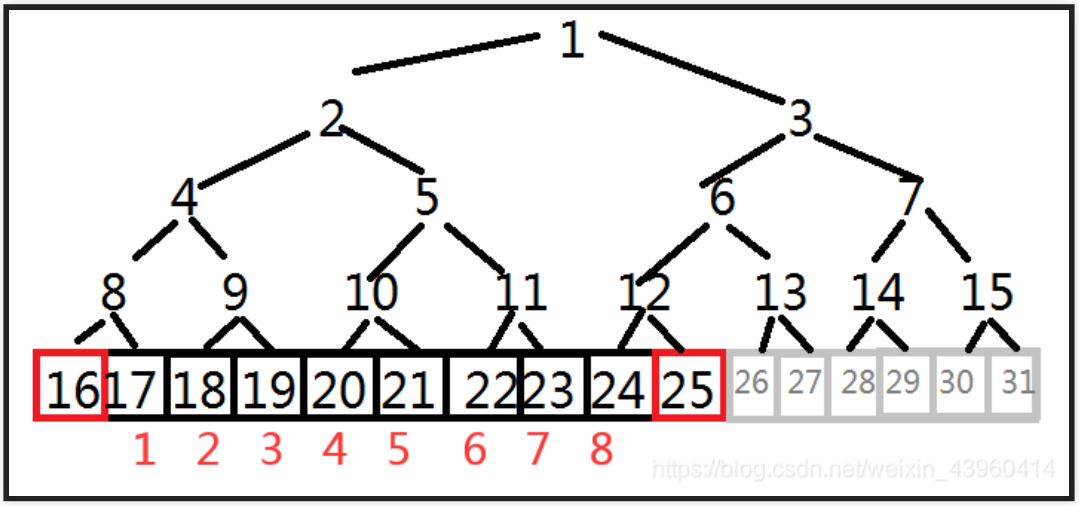
其他种类线段树

ZKW线段树

建树

因为是非递归的，所以是倒序建树。

ZKW是一颗满二叉树，大概是这个样子：



其他种类线段树

ZKW线段树

建树

可以发现，最下面一行的节点的深度，设为d，则有这棵树的总结点数是2^d，非叶子结点数为2^{d-1}，因此如果确定了d输入就很好处理了。需要注意的是，在查询的时候需要两个在查询区间外的节点，所以需要有冗余。（这个后面有讲，上图中红框中的节点就是查询时需要的冗余节点）

```
int M;//M为2^d
void build(int n){//叶子结点个数
    M = 1;
    while(M < 2 + n) //冗余
        M <<= 1;
    for(int i = M  - 1;i >= 1)tree[i] = opt(tree[i << 1],tree[i << 1 | 1]
}
```

其他种类线段树

ZKW线段树

单点修改

单点修改是先修改后代节点再更新父亲节点。

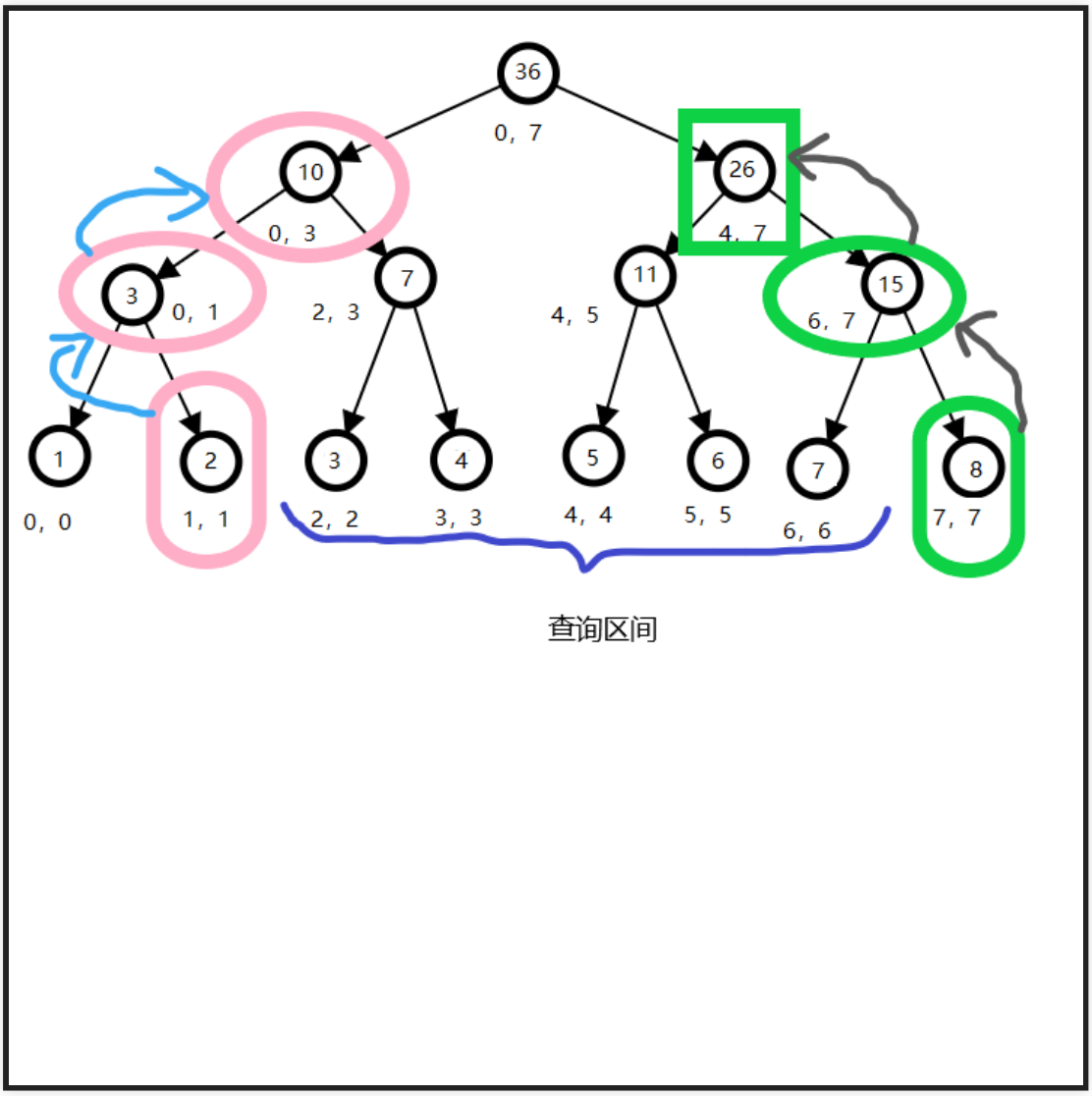
```
void modify(int x,int val){
    int p = M + x;
    tree[p] = val;
    while(p){
        tree[p >>= 1] = opt(tree[p >> 1],tree[p >> 1 | 1]);//根据实际情况而
    }
}
```

其他种类线段树

$\mathcal{ZKW}$线段树

区间查询

假设现在我们需要查询的是闭区间$[l,r]$，我们在区间外的两个点$M+(l-1)$和$M+(r+1)$建立两个指针s,t。



上图中粉色图形就是s的移动，绿色是t指针的移动。要查询的区间是大括号所包含的。两个指针s, t不断跳爸爸，直到两个指针指向的父节点相同为止。

跳的过程中，如果左指针指向的节点是左儿子，则$res+=$右儿子的值；如果右指针指向的节点是右儿

其他种类线段树

ZKW线段树

区间修改&标记永久化

因为ZKW线段树的查询和修改都是从下往上做的，所以普通线段树的懒标记下传是不好实现的，因此需要用到标记永久化（后面有讲）修改：

```
void modify(int l,int r,int val){
    int s = M + l - 1,t = M + r + 1;
    while(s || t){
        if(s < M)tree[s] = max(tree[s << 1],tree[s << 1 | 1]) + tag[s];
        if(t < M)tree[t] = max(tree[t << 1],tree[t << 1 | 1]) + tag[t];
        if((s >> 1) != (t >> 1)){
            if(!(s & 1))tree[s ^ 1] += val,tag[s ^ 1] += val;
            if(t & 1)tree[t ^ 1] += val,tag[t ^ 1] += val;
        }
        //不能break，因为tag标记要记录整个子树的修改
        s >>= 1;t >>= 1;
    }
    return ;
}
```

其他种类线段树

ZKW线段树

区间查询&标记永久化

```
int query(int l,int r){
    int s = M + l - 1,t = M + r + 1;
    int ls = 0,rs = 0;
    while(s || t){
        ls += tag[s];rs += tag[t];
        if((s >> 1) != (t >> 1)){
            if(!(s & 1))ls = max(ls,tree[s ^ 1]);
            if(t & 1)rs = max(rs,tree[t ^ 1]);
        }
        s >>= 1,t >>= 1;
    }
    return max(ls,rs);
}
```

标记永久化

简介

既然$\mathcal{ZKW}$用到了标记永久化，那么久顺便提一嘴。
其实主席树什么的都可以运用标记永久化。
顾名思义，标记永久化自然是不能下传的标记。主要用于比较复杂的，难以进行$pushdown$操作的数据结构，如主席树，树套树等。

标记永久化

实现

区间修改：

```
void modify(int x,int l,int r,int L,int R,int val){
    sum[x] += val * (r - l + 1);//对所有经过的区间直接修改
    if(l == L && r == R){tag[x] += v;return;}//对于完全覆盖的区间打上标记
    if(mid >= L)modify(ls,l,mid,L,R,val);
    if(mid < R)modify(rs,mid + 1,L,R,val);
}
```

标记永久化

实现

区间查询：

```
int query(int x,int l,int r,int L,int R,int add){
    if(l == L && R == r)return sum[x] + add * (r - l + 1);//注意这里，并没
    int res = 0;
    if(mid >= L)res += query(ls,l,mid,L,R,add + tag[x]);//要加上tag因为之前
    if(mid < R)res += query(rs,mid + 1,r,L,R,add + tag[x]);
    //注意在处理完tag后不能清零，因为可以注意到修改的时候并没有直接改变sum的值。
    return res;
}
```

标记永久化

例题

TO THE MOON

题面

一个长度为n的数组，4种操作：

$\mathcal{C}\ l\ r\ d$：区间$[l,r][l,r]$中的数都加d，同时当前的时间戳加1

$\mathcal{Q}\ l\ r$：查询当前时间戳区间$[l,r][l,r]$中所有数的和。

$\mathcal{H}\ l\ r\ t$：查询时间戳t区间$[l,r][l,r]$的和。

$\mathcal{B}\ t$：将当前时间戳置为t。

保证所有操作均合法。

标记永久化

例题

TO THE MOON

解析

主席树，然后我们显然没法下传标记，因为可能会影响到之前时间戳的状态，所以就标记永久化就好了。

标记永久化

例题

SP1741 TETRIS3D - TETRIS 3D

题面

有一个$D*S$平面，从上向下掉n个方块，直到碰到别的方块或地面才停下，问掉完后最高高度。

$1 \leq d, s \leq 1000, 1 \leq n \leq 20000$

标记永久化

例题

SP1741 TETRIS3D - TETRIS 3D

解析

线段树套线段树，一维表示横坐标，一维表示纵坐标，发现在更新外面一维的时候不方便更新，于是标记永久化。

标记永久化

例题

大家可以课后做一下这道题。

其他种类线段树

ZKW线段树

博客推荐

比较清楚，我看的是这个

洛谷日爆

还阔以

其他种类线段树

吉司机线段树

- 又称势能分析线段树

其他种类线段树

吉司机线段树

- 又称势能分析线段树

关于我打的暴力竟然不是暴力の这棵树

我的暴力线段树不可能这么快

暴力转生，上了树就拿出真速度~~

其他种类线段树

吉司机线段树

通常而言，这一类线段树看上去像是线段树套暴力，但是由于一些操作的优秀性质，你可以理性的证明它的复杂度是十分优异的。

其他种类线段树

吉司机线段树

第一个应用

给定一个序列，可以支持区间开根以及单点加减

其他种类线段树

吉司机线段树

第二个应用

给定一个序列，可以支持区间取模以及单点加减，求区间和

其他种类线段树

吉司机线段树

第三个应用

给定一个序列，可以支持区间取min值以及单点加减，求区间和

*区间取min值指把一个区间里的所有数替换成 $min(a_i, x)$

其他种类线段树

吉司机线段树

第三个应用

给定一个序列，可以支持区间取min值以及单点加减，求区间和

我们建线段树，维护区间最大值、次大值以及最大值数量，然后对于当前询问到的一个区间，分成下面三种情况：

对于$sec < max \leq x$的情况，我们显然不用替换

对于$sec \leq x < max$的情况，我们把所有最大值替换成x，然后改值就可以维护了

对于$x < sec < max$的情况，我们显然没有办法在当前这个区间处理，所以我们接着往下递归，知道碰到上面两种情况。

其他种类线段树

吉司机线段树

第三个应用

给定一个序列，可以支持区间取min值以及单点加减，求区间和

理性的复杂度分析（感谢ZHY花时间教会我这个FW）

把终止询问节点分成两类，一类是最大值与父亲节点相同的，一类是不同的。

一个中止节点的兄弟与它类别一定不同，且递归深度深于它

然后我们就可以把复杂度归到某一类上面去了

其他种类线段树

吉司机线段树

第三个应用

给定一个序列，可以支持区间取min值以及单点加减，求区间和

理性的复杂度分析

对于二类节点而言，每一次区间取min值操作都必定会减少至少一个二类节点。

每要到一个二类节点中止，就要递归$logn$层

没有二类节点显然不会递归

向下继续深入的复杂度$=$ 二类节点数量$logn$

其他种类线段树

吉司机线段树

第三个应用

给定一个序列，可以支持区间取min值以及单点加减，求区间和

注意到二类节点数量顶多是在n级别的，所以这个算法是在$O(nlog^2n)$级别的。

事实上还要快一点，接近$O(nlogn)$

其他种类线段树

吉司机线段树

[YZOJ4695]最假女选手

在刚刚结束的水题嘉年华的压轴节目放水大赛中，wyywyy如愿以偿的得到了最假女选手的奖项。但是作为主办人的
C_SUNSHINE为了证明wyywyy确实在放水，决定出一道基础题考察wyywyy的姿势水平。给定一个长度为 N 序列，编号
从1 到 N。要求支持下面几种操作：
1.给一个区间[L,R] 加上一个数x
2.把一个区间[L,R] 里小于x 的数变成x
3.把一个区间[L,R] 里大于x 的数变成x
4.求区间[L,R] 的和

其他种类线段树

吉司机线段树

[YZOJ4695]最假女选手

板子题，硬搞三个标记就好了，区间取min和区间取max的标记只会在区间长度小于等于2的时候才会互相影响。

分类讨论下，你先传哪个标记都没有影响。

其他种类线段树

吉司机线段树

[YZOJ4355]PLAY WITH SEQUENCE

维护一个长度为N的序列a，现在有三种操作：

1）给出参数U,V,C，将a[U],a[U+1],...,a[V-1],a[V]都赋值为C。

2）给出参数U,V,C，对于区间[U,V]里的每个数i，将a[i]赋值为max(a[i]+C,0)。

3）给出参数U,V，输出a[U],a[U+1],...,a[V-1],a[V]里值为0的数字个数。

其他种类线段树

吉司机线段树

[YZOJ4355]PLAY WITH SEQUENCE

把操作二改成先加后区间最值，然后套上操作一就好了

唯一要注意的地方是标记下传顺序，大概是先传区间覆盖，再传加减，最后是区间取最值

加减碰上取最值，改最值tag与加减tag，

最值碰上加减，只改最值tag，

最值碰上覆盖，改最值tag，

覆盖碰上最值，干掉最值tag，改覆盖tag，

加减碰上覆盖，改加减tag

覆盖碰上加减，干掉加减tag，修改覆盖tag

好吧写的很复杂其实你想怎么搞怎么搞，传tag的时候留个心眼保证先后顺序不错就好了。

其他种类线段树

吉司机线段树

[HDU6521]PARTY

有n个人，一开始都不认识彼此。现在举办m场party，每次参加的人是区间$[l_i, r_i]$内的人。参加完一次party之后，这区间内的人就会相互认识。问每次party后有多少人互相认识。

n和m都是 $5e5$ 级别。

其他种类线段树

吉司机线段树

[HDU6521]PARTY

别想其它复杂的操作然后势能分析了，就是区间取最值的直接运用

其他种类线段树

吉司机线段树

[HDU6521]PARTY

别想其它复杂的操作然后势能分析了，就是区间取最值的直接运用。

首先认识的人肯定是连续的

我们每个点维护一个L_i，表示这个人认识的最左边一个人是谁。然后注意到每一次party过后，所增加的朋友对数就是区间内所有人左区间拓展长度之和，也就是$\sum_{l_i \le x \le r_i} L_i - l + 1$

至于往右多认识的，会被右边的人统计到，所以其实我们维护个区间和就ok了，支持区间取最值操作。

THANKS ALL!