

扫描线问题

例 1

题意

P5490 【模板】扫描线

求 n 个矩形的面积并

题解

此问题可抽象出一个问题模型：

给一初始全 0 的序列 $\{a_i\}_{i=1}^n$ 每次操作如下：给定区间 $[l, r]$ 令 $a_i = a_i + 1, \forall i \in [l, r]$ 或 $a_i = a_i - 1, \forall i \in [l, r]$ 修改后立即询问 $\sum_{a_i > 0, i \in [1, n]} a_i$ 保证任一修改后 $a_i \geq 0, \forall i \in [1, n]$

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int MAXN=1e6+5;
#define ls o<<1
#define rs o<<1|1
ll X[MAXN<<1];
// 横线
struct Line{
    // 左右端点的横坐标,竖坐标
    ll l,r,h;
    // 1 表示下底边, -1 表示上底边
    int mark;
    // 按高度排序
    bool operator < (const Line &rhs) const {
        return h<rhs.h;
    }
}L[MAXN<<1];
// 线段树结点维护的信息
struct Info{
    // 区间左右端点,该区间位于多少个矩形内(值为正时有贡献)
    int l,r,sum;
    // 该区间的可做底边的长度(sum 值为正的长度)
    ll len;
};
struct SegmentTree{
    Info t[MAXN<<2];
    void Pull(int o){
        int l=t[o].l,r=t[o].r;
```

```
// 若非0 则有贡献
if(t[o].sum){
    t[o].len=X[r+1]-X[l];
}
else{
    t[o].len=t[ls].len+t[rs].len;
}
}

void Build(int o,int l,int r){
    t[o].l=l,t[o].r=r;
    t[o].len=0;
    t[o].sum=0;
    if(l==r){
        return;
    }
    int mid=(l+r)>>1;
    Build(ls,l,mid);
    Build(rs,mid+1,r);
    return;
}

void Update(int o,int nl,int nr,int c){
    int l=t[o].l,r=t[o].r;
    if(nl<=l&&r<=nr){
        t[o].sum+=c;
        Pull(o);
        return;
    }
    int mid=(l+r)>>1;
    if(nl<=mid)
        Update(ls,nl,nr,c);
    if(nr>mid)
        Update(rs,nl,nr,c);
    Pull(o);
}

}tree;

int main(){
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int n;
    cin>>n;
    for(int i=1;i<=n;i++){
        ll X1,X2,Y1,Y2;
        cin>>X1>>Y1>>X2>>Y2;
        X[2*i-1]=X1,X[2*i]=X2;
        L[2*i-1]=(Line){X1,X2,Y1,1};
        L[2*i]=(Line){X1,X2,Y2,-1};
    }
    n<<=1;
    sort(L+1,L+n+1);
    sort(X+1,X+n+1);
```

```
int tot=unique(X+1,X+n+1)-X-1;
tree.Build(1,1,tot-1);
ll ans=0;
for(int i=1;i<n;i++){
    int nl,nr;
    nl=lower_bound(X+1,X+tot+1,L[i].l)-X;
    nr=lower_bound(X+1,X+tot+1,L[i].r)-X-1;
    tree.Update(1,nl,nr,L[i].mark);
    ans+=tree.t[1].len*(L[i+1].h-L[i].h);
}

cout<<ans;
return 0;
}
```

From:

<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:

https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:lgwza:%E6%89%AB%E6%8F%8F%E7%BA%BF%E9%97%AE%E9%A2%98

Last update: 2021/09/17 11:17