

线段树题目整理

前话：之前写的线段树的代码不够优雅，所以这次准备开个坑重新写下代码，可能会咕咕咕掉。

1.P3372 【模板】线段树 1

区间修改，区间求和。

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1e5+5,M=1e6+5,inf=0x3f3f3f3f,mod=1e9+7;
#define mst(a) memset(a,0,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
struct node{
    int l,r,lz;
    ll s;
}a[N<<2];
void re(int x){
    a[x].s=a[lx].s+a[rx].s;
}
void pushdown(int x){
    if(a[x].lz){
        a[lx].lz+=a[x].lz,a[rx].lz+=a[x].lz;
        a[lx].s+=(a[lx].r-a[lx].l+1)*a[x].lz;
        a[rx].s+=(a[rx].r-a[rx].l+1)*a[x].lz;
        a[x].lz=0;
    }
}
void build(int x,int l,int r){
    a[x].l=l,a[x].r=r;
    if(l==r){
        scanf("%lld",&a[x].s);
        return;
    }
    int mid=(l+r)>>1;
    build(lx,l,mid);
    build(rx,mid+1,r);
    re(x);
}
void update(int x,int l,int r,int val){
    if(a[x].l>=l&&a[x].r<=r){
        a[x].s+=1LL*(a[x].r-a[x].l+1)*val;
        a[x].lz+=val;return;
    }
    pushdown(x);
    int mid=(a[x].l+a[x].r)>>1;
    if(l<=mid) update(lx,l,r,val);
```

```

        if(r>mid) update(rx,l,r,val);
        re(x);
    }
    ll query(int x,int l,int r){
        if(a[x].l>=l&&a[x].r<=r) return a[x].s;
        int mid=(a[x].l+a[x].r)>>1;
        pushdown(x);
        ll ans=0;
        if(l<=mid) ans+=query(lx,l,r);
        if(r>mid) ans+=query(rx,l,r);
        return ans;
    }
    int main(){
        int n,m;
        scanf("%d%d",&n,&m);
        build(1,1,n);
        while(m--){
            int op,l,r,k;
            scanf("%d%d%d",&op,&l,&r);
            if(op==1) scanf("%d",&k),update(1,l,r,k);
            else printf("%lld\n",query(1,l,r));
        }
        return 0;
    }
}

```

2.P3373 【模板】线段树 2

区间乘 x ，区间加 x ，区间求和。

按照先乘后加原则更新 $lazytag$ 即可。

因为 $(x+2) \times 3 + 4 = x \times 3 + 2 \times 3 + 4$ 。

而先加后乘会出现 $(x+2+\frac{4}{3}) \times 3$ 含浮点数计算。

具体看代码。

```

#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1e5+5,M=1e6+5,inf=0x3f3f3f3f;
#define mst(a) memset(a,0,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
int mod;
struct node{
    int l,r;
    ll lz,mt,s;//lz加法标记,mt(multi)乘法标记
}a[N<<2];
void re(int x){
    a[x].s=(a[lx].s+a[rx].s)%mod;
}
void pushdown(int x){

```

```

a[lx].s=(a[lx].s*a[x].mt+a[x].lz*(a[lx].r-a[lx].l+1))%mod; //先乘后加.
a[rx].s=(a[rx].s*a[x].mt+a[x].lz*(a[rx].r-a[rx].l+1))%mod;
a[lx].mt=a[lx].mt*a[x].mt%mod;
a[rx].mt=a[rx].mt*a[x].mt%mod;
a[lx].lz=(a[lx].lz*a[x].mt+a[x].lz)%mod; //这里加法lazy_tag 也要先乘后加.
a[rx].lz=(a[rx].lz*a[x].mt+a[x].lz)%mod;
a[x].lz=0,a[x].mt=1;//还原.
}
void build(int x,int l,int r){
    a[x].l=l,a[x].r=r,a[x].mt=1;
    if(l==r){
        scanf("%lld",&a[x].s);
        return;
    }
    int mid=(l+r)>>1;
    build(lx,l,mid);
    build(rx,mid+1,r);
    re(x);
}
void update(int x,int l,int r,int op,ll val){
    if(a[x].l>=l&&a[x].r<=r){
        if(op==1){
            a[x].mt=a[x].mt*val%mod;
            a[x].lz=a[x].lz*val%mod; //这里加法也要更新.
            a[x].s=a[x].s*val%mod;
        }
        else {
            a[x].lz=(a[x].lz+val)%mod;
            a[x].s=(a[x].s+val*(a[x].r-a[x].l+1))%mod;
        }
        return;
    }
    pushdown(x);
    int mid=(a[x].l+a[x].r)>>1;
    if(l<=mid) update(lx,l,r,op,val);
    if(r>mid) update(rx,l,r,op,val);
    re(x);
}
ll query(int x,int l,int r){
    if(a[x].l>=l&&a[x].r<=r) return a[x].s;
    int mid=(a[x].l+a[x].r)>>1;
    pushdown(x);
    ll ans=0;
    if(l<=mid) ans=(ans+query(lx,l,r))%mod;
    if(r>mid) ans=(ans+query(rx,l,r))%mod;
    return ans;
}
int main(){
    int n,m;
    scanf("%d%d%d",&n,&m,&mod);
    build(1,1,n);
    while(m--){
        int op,l,r,k;
        scanf("%d%d%d",&op,&l,&r);
        if(op==3) printf("%lld\n",query(1,l,r));
        else scanf("%d",&k),update(1,l,r,op,k);
    }
    return 0;
}

```

```
}
```

3.P5490 【模板】扫描线

求 n 个矩形的面积并。

思路：板子题，具体见代码。

坑点：数组要开大，否则可能会 *RE*，开4倍都会炸，要开8倍。

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=4e5+5,M=1e6+5,inf=0x3f3f3f3f,mod=1e9+7;
#define mst(a) memset(a,0,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
ll x[N<<1];
struct node{
    ll l,r,s,len;//l,r,s,len 结点区间表示线段[x[l],x[r+1]], s表示线段的个数,和线段长度.
}a[N<<2];
struct Line{    //储存所有矩形横边,从下往上扫描.
    ll l,r,h,c; //l,r,h,c分别为横线左端点,右端点,距离x轴高度,下边权值c=1,上边权值
    c=-1>
    bool operator<(const Line&li)const{
        return h<li.h;
    }
}line[N<<1];
void build(int x,int l,int r){ //建树
    a[x].l=l,a[x].r=r;
    if(l==r) return;
    int mid=(l+r)>>1;
    build(lx,l,mid);
    build(rx,mid+1,r);
}
void pushup(int x){ //更新.
    int l=a[x].l,r=a[x].r;
    if(a[x].s) a[x].len=x[r+1]-x[l]; //如果当前线段已经被记数了,直接取长度
    else a[x].len=a[lx].len+a[rx].len;//否则更新.
}
void update(int x,int L,int R,int val){ //更新线段记数情况.
    int l=a[x].l,r=a[x].r;
    if(x[r+1]<=L||x[l]>=R) return;
    if(x[l]>=L&& x[r+1]<=R){
        a[x].s+=val;
        pushup(x);
        return;
    }
    update(lx,L,R,val);
    update(rx,L,R,val);
    pushup(x);
}
int main(){
```

```

int n,cnt=0;
scanf("%d",&n);
for(int i=1;i<=n;i++){
    int x1,y1,x2,y2;
    scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
    x[(i<<1)-1]=x1,x[(i<<1)]=x2;
    line[(i<<1)-1]={x1,x2,y1,1};
    line[(i<<1)]={x1,x2,y2,-1};
}
n<<=1;
sort(line+1,line+n+1);
sort(X+1,X+n+1);
cnt=unique(X+1,X+n+1)-X-1; //离散化所有端点的横坐标x.从小到大排序.
build(1,1,cnt-1);
ll ans=0;
for(int i=1;i<n;i++){ //<n 不用计算最后一条横边.
    update(1,line[i].l,line[i].r,line[i].c);
    ans+=a[1].len*(line[i+1].h-line[i].h);
}
printf("%lld\n",ans);
return 0;
}

```

4.P4145 上帝造题的七分钟2 / 花神游历各国

区间开平方，区间求和。

我们用一个标记记录当前区间最大的数。

显然 10^{12} 开6次就到了1.5了，所以暴力修改就行了。

```

#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int N=1e5+5,M=1e6+5,inf=0x3f3f3f3f;
#define mst(a) memset(a,0,sizeof a)
#define lx x<<1
#define rx x<<1|1
#define reg register
#define PII pair<int,int>
#define fi first
#define se second
int mod;
struct node{
    int l,r;
    ll lz,mx,s;
}a[N<<2];
template<class T>
inline void read(T &x){
    x=0;int w=1;
    char ch=getchar();
    while(ch<'0' || ch>'9') {if(ch=='-') w=-1;ch=getchar();}
    for(;ch>='0'&&ch<='9';ch=getchar())
        x=(x<<3)+(x<<1)+(ch&15);
    x*=w;
}
void re(int x){
    a[x].s=a[lx].s+a[rx].s;
}

```

```

        if(a[lx].mx>a[rx].mx) a[x].mx=a[lx].mx;
        else a[x].mx=a[rx].mx;
    }
    void build(int x,int l,int r){
        a[x].l=l,a[x].r=r;
        if(l==r){
            read(a[x].s),a[x].mx=a[x].s;
            return;
        }
        int mid=(l+r)>>1;
        build(lx,l,mid);
        build(rx,mid+1,r);
        re(x);
    }
    void update(int x,int l,int r){
        if(a[x].l==a[x].r){
            a[x].s=sqrt(a[x].s);
            a[x].mx=sqrt(a[x].mx);
            return ;
        }
        int mid=(a[x].l+a[x].r)>>1;
        if(l<=mid&& a[lx].mx>1) update(lx,l,r);
        if(r>mid&&a[rx].mx>1) update(rx,l,r);
        re(x);
    }
    ll query(int x,int l,int r){
        if(a[x].l>=l&&a[x].r<=r) return a[x].s;
        int mid=(a[x].l+a[x].r)>>1;
        ll ans=0;
        if(l<=mid) ans+=query(lx,l,r);
        if(r>mid) ans+=query(rx,l,r);
        return ans;
    }
    int main(){
        int n,m;
        read(n);
        build(1,1,n);
        read(m);
        while(m--){
            int op,l,r;
            read(op),read(l),read(r);
            if(l>r) swap(l,r);
            if(op) printf("%lld\n",query(1,l,r));
            else update(1,l,r);
        }
        return 0;
    }
}

```