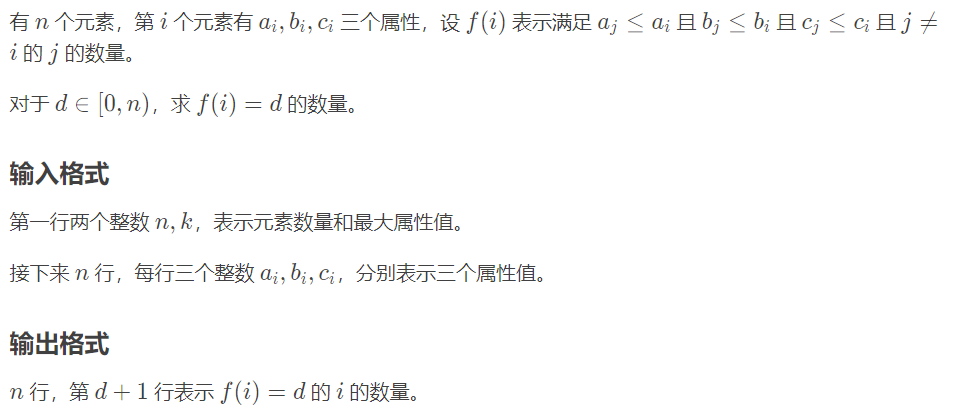
P3810 【模板】三维偏序（陌上花开）



#include<bits/stdc++.h>

#define ll long long

#define inf 1000000000000000000

#define maxn 1000005

#define bit(i) (i&(-i))

using namespace std;

ll c[maxn],ans[maxn]={0},maxx,minn,cnt=0,tot=0,ge,n,m;

struct w{ll a,b,c,cnt,ans;}a[maxn],p[maxn];

bool cmp(const w&a,const w&b)

{

return a.a<b.a||a.a==b.a&&a.b<b.b||a.a==b.a&&a.b==b.b&&a.c<b.c;

}

bool cmp2(const w&a,const w&b)

{

return a.b<b.b||a.b==b.b&&a.c<b.c;

}

void add(ll x,ll y)

{ll i;

for(i=x;i<=m;i+=bit(i))

c[i]+=y;

}

ll ask(ll x)

{

ll ans=0,i;

for(i=x;i;i-=bit(i))

ans+=c[i];

return ans;

}

void cdq(ll l,ll r)

{

if(l==r){return ;}

ll mid=(l+r)/2;

cdq(l,mid);

cdq(mid+1,r);

sort(p+l,p+mid+1,cmp2);

sort(p+mid+1,p+r+1,cmp2);

ll sum=0;

ll i=l,j=mid+1,o=l;

while(i<=mid&&j<=r)

{

if(p[i].b<=p[j].b){add(p[i].c,p[i].cnt);i++;}

else {p[j].ans+=ask(p[j].c);j++;}

}

while(j<=r){p[j].ans+=ask(p[j].c);j++;}

for(j=l;j<i;j++)

add(p[j].c,-p[j].cnt);

}

int main()

{

ll num=0,h,f1,f2,x,y,k,len,t,i,j,z;

char ch[10];

cin>>n>>m;

for(i=1;i<=n;i++)

scanf("%lld%lld%lld",&a[i].a,&a[i].b,&a[i].c);

sort(a+1,a+n+1,cmp);

for(i=1;i<=n;i++)

{

cnt++;

if(a[i].a!=a[i+1].a||a[i].b!=a[i+1].b||a[i].c!=a[i+1].c)

{

p[++tot]=a[i];

p[tot].cnt=cnt;

cnt=0;

}

}

cdq(1,tot);

for(i=1;i<=tot;i++)

ans[p[i].ans+p[i].cnt-1]+=p[i].cnt;

for(i=0;i<=n-1;i++)

printf("%lld\n",ans[i]);

return 0;

}

其他人的版本

#include<cstdio>

#include<algorithm>

#define maxn 200005

using namespace std;

struct node

{

int a,b,c,cnt,ans;

}s1[maxn],s2[maxn];

int n,m,k,mx,top,su[maxn];

int c[maxn];//树状数组

bool cmp1(node x,node y)

{

if(x.a==y.a)

{

if(x.b==y.b)return x.c<y.c;

else return x.b<y.b;

}

else return x.a<y.a;

}//第一维排序

bool cmp2(node x,node y)

{

if(x.b==y.b)

return x.c<y.c;

else return x.b<y.b;

}//第二维排序

int lowbit(int x)

{

return x&(-x);

}

void add(int x,int y)

{

while(x<=mx)

{

c[x]+=y;

x+=lowbit(x);

}

}//树状数组单点加

int query(int x)

{

int sum=0;

while(x)

{

sum+=c[x];

x-=lowbit(x);

}

return sum;

}//求单点前缀和

//树状数组看得懂吧QAQ

void cdq(int l,int r)

{

if(l==r)return;

int mid=(l+r)>>1;

cdq(l,mid);

cdq(mid+1,r);//类似于归并排序

sort(s2+l,s2+mid+1,cmp2);

sort(s2+mid+1,s2+r+1,cmp2);//第二维为关键字排序

int i,j=l;

for(i=mid+1;i<=r;++i)

{

while(s2[i].b>=s2[j].b&&j<=mid)

{

add(s2[j].c,s2[j].cnt);//在s2[j]位置加上s2[j]的个数

j++;

}

s2[i].ans+=query(s2[i].c);//保证树状数组里的数一定符合条件

}//类似归并

for(i=l;i<j;++i)

add(s2[i].c,-s2[i].cnt);//清空树状数组

}//cdq分治

int main()

{

scanf("%d%d",&n,&k);

mx=k;//树状数组的区间

for(int i=1;i<=n;++i)

{

int a,b,c;

scanf("%d%d%d",&a,&b,&c);

s1[i].a=a;

s1[i].b=b;

s1[i].c=c;

}//初始化输入

sort(s1+1,s1+1+n,cmp1);//第一维为关键字排序

for(int i=1;i<=n;++i)

{

top++;

if(s1[i].a!=s1[i+1].a||s1[i].b!=s1[i+1].b||s1[i].c!=s1[i+1].c)

{

m++;

s2[m].a=s1[i].a;

s2[m].b=s1[i].b;

s2[m].c=s1[i].c;

s2[m].cnt=top;

top=0;

}

}//第一维已有序,合并相同节点

cdq(1,m);//cdq分治

for(int i=1;i<=m;++i)

su[s2[i].ans+s2[i].cnt-1]+=s2[i].cnt;

for(int i=0;i<n;++i)

printf("%d\n",su[i]);

return 0;

}

