线段树 区间更新

懒标记不是加上了，而是修改为这个值

#include <cstdio>

#include<iostream>

#include<vector>

#include<algorithm>

#include<string>

#include<cstring>

#include<set>

#include<map>

using namespace std;

#define ll long long

#define INF 1000000007

int a[100010\*4];

int lazy[100010\*4];

int len[100010\*4];

int x,y;

int z;

inline void build(int root,int l,int r)

{

len[root]=r-l+1;

if(l==r)

{

a[root]=1;

return ;

}

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

build(root<<1,l,mid);

build(root<<1|1,mid+1,r);

a[root]=a[root<<1]+a[root<<1|1];

}

inline void pushdown(int root)

{

if(!lazy[root])return;

lazy[root<<1]=lazy[root];

lazy[root<<1|1]=lazy[root];

a[root<<1]=len[root<<1]\*lazy[root];

a[root<<1|1]=len[root<<1|1]\*lazy[root];

lazy[root]=0;

}

inline void update(int x,int y,int root,int l,int r)

{

if(x<=l && r<=y)//单点更新的时候是if(l==r)return a[root]，但这是区间更新

{

a[root]=len[root]\*z;

lazy[root]=z;

return;

}

pushdown(root);//只有在需要计算子区间的值的时候才将标记下传

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

if(y<=mid)//要找的区间整个在mid左侧

update(x,y,root<<1,l,mid);

else if(x>mid)//要找的区间整个在mid右侧

update(x,y,root<<1|1,mid+1,r);

else

{

update(x,mid,root<<1,l,mid);

update(mid+1,y,root<<1|1,mid+1,r);

}

a[root]=a[root<<1]+a[root<<1|1];

}

inline int getsum(int x,int y,int root,int l,int r)

{

if(x<=l && r<=y)

return a[root];

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

pushdown(root);//只有在需要计算子区间的值的时候才将标记下传

//一次只是下传一层

int ans=0;

if(x<=mid)//中间的mid点还是在左区间上

ans+=getsum(x,y,root<<1,l,mid);

if(y>mid)

ans+=getsum(x,y,root<<1|1,mid+1,r);

return ans;

}

int main()

{

//freopen("input.txt","r",stdin);

int n,q,T;

scanf("%d",&T);

for(int cas=1;cas<=T;++cas)

{

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

memset(lazy,0,sizeof(lazy));

memset(len,0,sizeof(len));

build(1,1,n);

while(q--)

{

scanf("%d%d%d",&x,&y,&z);

update(x,y,1,1,n);

}

printf("Case %d: The total value of the hook is %d.\n",cas,getsum(1,n,1,1,n));

}

return 0;

}