

线段树区间更改和区间求和

```
#include <bits/stdc++.h>
#define lid id<<1
#define rid id<<1|1

#define INF 0x3f3f3f3f
#define maxn 100005
typedef long long ll;
using namespace std;

int s[maxn]; // 储存节点值
ll ans; // 区间值总和，每次查询清零
struct node
{
    int l, r; // 左右端点
    ll sum, add; // 总和及该区间需要更新的多少
} tr[maxn*4]; // 树

void push_up(int id) // 上传子节点状态
{
    tr[id].sum = tr[lid].sum + tr[rid].sum;
}

void build(int id, int l, int r) // 建树
{
    tr[id].l = l; tr[id].r = r;
    tr[id].add = 0;
    if (l == r)
    {
        tr[id].sum = s[l];
        return;
    }
    int mid = (l + r) >> 1;
    build(lid, l, mid);
    build(rid, mid + 1, r);
    push_up(id);
}

void update(int id, int l, int r, int add) // 更新给定区间全部值
{
    if (l > tr[id].r || r < tr[id].l) return;
    if (l <= tr[id].l && r >= tr[id].r)
    {
        tr[id].sum += (tr[id].r - tr[id].l + 1) * add;
        tr[id].add += add;
        return;
    }
    if (tr[id].add) // 传递状态给子节点
    {
```

```

        tr[lid].sum+=(tr[lid].r-tr[lid].l+1)*tr[id].add;
        tr[lid].add+=tr[id].add;
        tr[rid].sum+=(tr[rid].r-tr[rid].l+1)*tr[id].add;
        tr[rid].add+=tr[id].add;
        tr[id].add=0;
    }
    update(lid,l,r,add);
    update(rid,l,r,add);
    push_up(id);
}
void query(int id,int l,int r)//查询区间总和
{
    if(tr[id].r<l||tr[id].l>r) return ;
    if(tr[id].l>=l&&tr[id].r<=r)
    {
        ans+=tr[id].sum;
        return ;
    }
    if(tr[id].add)//传递状态给子节点
    {
        tr[lid].sum+=(tr[lid].r-tr[lid].l+1)*tr[id].add;
        tr[lid].add+=tr[id].add;
        tr[rid].sum+=(tr[rid].r-tr[rid].l+1)*tr[id].add;
        tr[rid].add+=tr[id].add;
        tr[id].add=0;
    }
    query(lid,l,r);
    query(rid,l,r);
    push_up(id);
}

int main()
{
    int n,q;
    scanf("%d%d",&n,&q);
    for(int i=1;i<=n;i++) {
        scanf("%d",&s[i]);
    }
    build(1,1,n);
    while(q--)
    {
        getchar();
        char ch;
        scanf("%c",&ch);
        if(ch=='Q')//询问
        {
            int l,r;
            ans=0;
            scanf("%d%d",&l,&r);

```

```
        query(1,l,r);
        printf("%lld\n",ans);
    }
    else//更新
    {
        int l,r,add;
        scanf("%d%d%d",&l,&r,&add);
        update(1,l,r,add);
    }
}
return 0;
}
```