树状数组单点更新和区间求和

```
#include <cstdio>
#include <stack>
#include <cmath>
#include <queue>
#include <string>
#include <queue>
#include <cstring>
#include <iostream>
#include <algorithm>

#define lid id<<1
#define rid id<<1|1
#define closein cin.tie(0)
#define scac(a) scanf("%c",&a)
#define scad(a) scanf("%d",&a)
#define print(a) printf("%d\n",a)
#define debug printf("hello world")
#define form(i,n,m) for(int i=n;i<m;i++)
#define mfor(i,n,m) for(int i=n;i>m;i--)
#define nfor(i,n,m) for(int i=n;i>=m;i--)
#define forn(i,n,m) for(int i=n;i<=m;i++)
#define scadd(a,b) scanf("%d%d",&a,&b)
#define memset0(a) memset(a,0,sizeof(a))
#define scaddd(a,b,c) scanf("%d%d%d",&a,&b,&c)
#define scadddd(a,b,c,d) scanf("%d%d%d%d",&a,&b,&c,&d)

#define INF 0x3f3f3f3f
#define maxn 50005
typedef long long ll;
using namespace std;
//---------AC(^-^)AC---------\\

int n;
int tree[maxn];

int lowbit(int x)
{
        return x&(-x);
}
void update(int i,int add)//单点更新
{
        for(int j=i;j<=n;j+=lowbit(j))
        {
                tree[j]+=add;
        }
}
```

```
int query(int l,int r)//区间求和
{
        int ans1=0;
        for(int i=r;i>0;i-=lowbit(i))
        {
                ans1+=tree[i];
        }
        int ans2=0;
        for(int i=l-1;i>0;i-=lowbit(i))
        {
                ans2+=tree[i];
        }
        return ans1-ans2;
}

int main()
{
        int T;
        int cnt=0;
        scanf("%d",&T);
        while(T--)
        {
                memset(tree,0,sizeof(tree));//清空数组
                scanf("%d",&n);
                for(int i=1;i<=n;i++)
                {
                        int m;
                        scanf("%d",&m);
                        update(i,m);
                }
                char s[10];

                while(scanf("%s",s)!=EOF)
                {
                        if(s[0]=='A')   {
                                int i,j;
                                scanf("%d%d",&i,&j);
                                update(i,j);
                        }
                        else if(s[0]=='S')
                        {
                                int i,j;
                                scanf("%d%d",&i,&j);
                                update(i,-j);
                        }
                        else if(s[0]=='Q')
                        {
                                int i,j;
                                scanf("%d%d",&i,&j);
```

```
                        printf("%d\n",query(i,j));
                    }
                    else break;
                }
        }
        return 0;
}
```