

题意

线段树单点修改，区间查询 模板题

思路

三个模块

1. void build

建树的时候，向下递归先取左区间再取右区间，访问到叶子结点的先后顺序就是n个数的顺序，所以可以遇到叶子结点再输入

$O(n\log n)$

2. int query

将当前区间[l,r] 从中间分成左右部分[l,m] [m+1,r] 如果某部分与待查询区间[L,R]无交集，则不向下查询，视作该部分的返回值为0

当前区间的返回值 = 左返回值 + 右返回值

3. void add

因为单点修改，所以每次二分为两部分后，待修改点x，只会存在于其中一个区间内，所以每次舍弃一半区间

从根节点，到待修改点x，所经历的每一个结点，均加上修改值d

代码

```
1  #include<iostream>
2  #define For(i,a,b) for(int i=(a); i<=(b) ; i++)
3  #define _For(i,a,b) for(int i=(a); i>=(b) ; i--)
4  #define Memset(a,b); memset((a),(b),sizeof((a)));
5  #define Cin(a); scanf("%d",&(a));
6  #define Cinc(a); scanf(" %c",&(a));
7  #define Cins(a); scanf("%s",(a));
8  #define Cout(a,b); printf("%d",(a));printf(b);
9  #define Coutc(a,b); printf("%c",(a));printf(b);
10 #define Couts(a,b); printf("%s",(a));printf(b);
11 using namespace std;
12 typedef long long LL;
13 typedef unsigned long long ULL;
14 typedef long double LDB;
15 inline int readint() {int x;cin>>x;return x;}
16 int tree[200005];
17 void build(int o,int l,int r)
```

```

18 {
19     if(l == r){
20         tree[o] = readint();return ;
21     }
22     int M = ((l+r)>>1);
23     build(o<<1,l,M); build(o<<1|1,M+1,r);
24     tree[o] = tree[o<<1]+tree[o<<1|1];
25     return ;
26 }
27 int query(int o,int l,int r,int L,int R)
28 {
29     if(l>=L && r<=R) return tree[o];
30     int M = ((l+r)>>1),sum = 0;
31     if(M>=L) sum+=query(o<<1,l,M,L,R);
32     if(M+1<=R) sum+=query(o<<1|1,M+1,r,L,R);
33     return sum;
34 }
35 void add(int o,int l,int r,int x,int d)
36 {
37     if(l == r) {
38         tree[o]+=d;return;
39     }
40     int M = (l+r)>>1;
41     if(M>=x) add(o<<1,l,M,x,d);
42     else add(o<<1|1,M+1,r,x,d);
43     tree[o] += d;
44     return ;
45 }
46 int n;
47 char cmd[10];
48 int x,y;
49
50 int main()
51 {
52     int T;
53     Cin(T);
54     For(_,1,T)
55     {
56         Cin(n);
57         build(1,1,n);
58         printf("Case %d:\n",_);
59         while(1)
60         {
61             Cins(cmd);
62             if(cmd[0] == 'E') break;
63             Cin(x);Cin(y);

```

```
64         if(cmd[0] == 'Q') {
65             int ans = query(1,1,n,x,y);
66             Cout(ans,"\n");
67         }
68         if(cmd[0] == 'A') add(1,1,n,x,y);
69         if(cmd[0] == 'S') add(1,1,n,x,-y);
70     }
71 }
72 }
```