

题意

给出一个学生成绩的序列，有两个操作：

- 1.修改一个学生的成绩，
- 2.查询学生A到学生B之间所有学生的最高分。

思路

单点更新，求区间最大值，直接套用模板

HDU关同步然并卵

最开始还是得建树，直接依次插入会TLE

代码

上交红书

```
1  #include<map>
2  #include<set>
3  #include<stack>
4  #include<queue>
5  #include<cmath>
6  #include<ctime>
7  #include<vector>
8  #include<cstdio>
9  #include<cctype>
10 #include<cstring>
11 #include<cstdlib>
12 #include<iostream>
13 #include<algorithm>
14 using namespace std;
15 typedef long long LL;
16 #define T_S (1<<(20))
17 int n,m;
18 int aa[200005];
19 class IntervalTree{
```

```

20     private:
21         int Cover[T_S],Top[T_S];
22         int size;
23         void _Build(int l,int r,int id)
24         {
25             if(l == r)
26             {
27                 Top[id] = aa[l];
28                 return ;
29             }
30             int mid = (l+r)/2;
31             _Build(l,mid,id<<1);
32             _Build(mid+1,r,(id<<1)|1);
33             Top[id] = max(Top[id<<1],Top[(id<<1)|1]);
34         }
35         int _Query(int a,int b,int l,int r,int Ind)
36         {
37             if(a<=l &&b>=r) return Top[Ind];
38             int mid = (l+r)>>1,ret = Cover[Ind];
39             if(a<=mid) ret = max(ret , _Query(a,b,l,mid,Ind<<1));
40             if(b>mid) ret = max(ret,_Query(a,b,mid+1,r,(Ind<<1)|1));
41             return ret;
42         }
43         void _Modify(int a,int l,int r,int id,int d)
44         {
45             if(l==r&&l==a)
46             {
47                 Cover[id] = Top[id] = d;
48                 return;
49             }
50             if(a<l || a> r)
51             {
52                 return ;
53             }
54             int mid = (l+r) >>1;
55             if(a<=mid) _Modify(a,l,mid,id<<1,d);
56             else _Modify(a,mid+1,r,(id<<1)|1,d);
57             Top[id] = max(Top[id<<1],Top[(id<<1)|1]);
58         }
59     public:
60         IntervalTree()
61         {
62             memset(Cover,0,sizeof(Cover));
63             memset(Top,0,sizeof(Top));
64         }
65         void init(int nn)

```

```

66     {
67         memset(Cover,0,sizeof(Cover));
68         memset(Top,0,sizeof(Top));
69         size = n;
70     }
71     int Query(int a,int b)
72     {
73         return _Query(a,b,1,size,1);
74     }
75     void Modify(int a,int d)
76     {
77         return _Modify(a,1,size,1,d);
78     }
79     void Build()
80     {
81         return _Build(1,size,1);
82     }
83 };
84 IntervalTree tree;
85 int main()
86 {
87     // ios::sync_with_stdio(false);
88     while(scanf("%d%d",&n,&m)!=EOF)
89     {
90         tree.init(n);
91         for(int i=1;i<=n;i++)
92         {
93             scanf("%d",&aa[i]);
94         }
95         tree.Build();
96         char c;
97         int a,b;
98         for(int i =1;i<=m;i++)
99         {
100             scanf(" %c",&c);
101             scanf("%d%d",&a,&b);
102             if(c == 'Q')
103             {
104                 printf("%d\n",tree.Query(a,b));
105                 //cout<<tree.Query(a,b)<<endl;
106             }
107             else if (c == 'U')
108             {
109                 tree.Modify(a,b);
110             }
111         }

```

```
112     }
113 }
```

自己的

```
1  #include<iostream>
2  #include<cstring>
3  #include<algorithm>
4  #define For(i,a,b) for(int i=(a); i<=(b) ; i++)
5  #define _For(i,a,b) for(int i=(a); i>=(b) ; i--)
6  #define Memset(a,b); memset((a),(b),sizeof((a)));
7  #define Cin(a); scanf("%d",&(a));
8  #define Cinc(a); scanf(" %c",&(a));
9  #define Cins(a); scanf("%s",(a));
10 #define Cout(a,b); printf("%d",(a));printf(b);
11 #define Coutc(a,b); printf("%c",(a));printf(b);
12 #define Couts(a,b); printf("%s",(a));printf(b);
13 using namespace std;
14 typedef long long LL;
15 typedef unsigned long long ULL;
16 typedef long double LDB;
17 inline int readint() {int x;cin>>x;return x;}
18 int tree[1000005];
19 void build(int o,int l,int r)
20 {
21     if(l == r){
22         tree[o] = readint();return ;
23     }
24     int M = ((l+r)>>1);
25     build(o<<1,l,M); build(o<<1|1,M+1,r);
26     tree[o] = max(tree[o<<1],tree[o<<1|1]);
27     return ;
28 }
29 int query(int o,int l,int r,int L,int R)
30 {
31     if(l>=L && r<=R) return tree[o];
32     int M = ((l+r)>>1);
33     if(M>=L && M+1<=R){
34         int k1 = query(o<<1,l,M,L,R);
35         int k2 = query(o<<1|1,M+1,r,L,R);
36         return max(k1,k2);
37     }
38     else if(M>=L){
39         int k = query(o<<1,l,M,L,R);
40         return k;
41     }
42     else if(M+1<=R){
43         int k = query(o<<1|1,M+1,r,L,R);
44         return k;
45     }
46 }
```

```

41     }
42     else if(M+1<=R) {
43         int k = query(o<<1|1,M+1,r,L,R);
44         return k;
45     }
46 }
47 void update(int o,int l,int r,int x,int d)
48 {
49     if(l == r) {
50         tree[o]=d;return;
51     }
52     int M = (l+r)>>1;
53     if(M>=x)
54     {
55         update(o<<1,l,M,x,d);
56         tree[o] = max(tree[o<<1],tree[o<<1|1]);
57     }
58     else
59     {
60         update(o<<1|1,M+1,r,x,d);
61         tree[o] = max(tree[o<<1],tree[o<<1|1]);
62     }
63     return ;
64 }
65 int n,m;
66 char cmd;
67 int x,y;
68 int main()
69 {
70     while(scanf("%d%d",&n,&m)!=EOF)
71     {
72         Memset(tree,0);
73         build(1,1,n);
74         For(i,1,m)
75         {
76             Cinc(cmd);Cin(x);Cin(y);
77             if(cmd == 'Q') {
78                 int ans = query(1,1,n,x,y);
79                 Cout(ans,"\n");
80             }
81             if(cmd == 'U') update(1,1,n,x,y);
82         }
83     }
84 }

```