

目录

1	区间查询，统计两个相同概率	1
2	时间戳 + 统计有多少个不同的数	3
3	树状数组维护区间两数之差	4
4	统计有多少个不同的数	5

1 区间查询，统计两个相同概率

```

1  /*
2     HYSBZ - 2038 小Z的袜子
3     统计[L,R]区间内选两只袜子，颜色相同的概率。区间查询，统计两个相同概率。
4      $N, M \leq 50000, 1 \leq L < R \leq N, C_i \leq N$ 。
5  */
6  #define ll long long
7  const ll SIZE=50050;
8  struct node{//查询
9      ll L,R;
10     ll id,res1,res2;
11 }q[SIZE];
12
13 ll c[SIZE];//n只袜子的颜色
14 ll n,m,unit;//n只袜子，m次查询
15 ll num[SIZE];
16
17 ll com(ll nn,ll mm){
18     if(mm>nn/2)mm=nn-mm;
19     ll aa=1,bb=1;
20     for(ll i=1;i<=mm;i++){
21         aa*=nn+1-i;
22         bb*=i;
23         if(aa%bb==0){
24             aa/=bb;
25             bb=1;
26         }
27     }
28     ll tmp=aa/bb;
29     return tmp;
30 }
31

```

```

32 //朴素排序
33 bool cmp(node a,node b){
34     if(a.L/unit != b.L/unit)return a.L/unit < b.L/unit;
35     else return a.R < b.R;
36 }
37
38 //对答案进行排序
39 bool cmp_id(node a,node b){
40     return a.id<b.id;
41 }
42
43 int main(){
44     cin>>n>>m;
45     unit=sqrt(n); //分块
46     for(ll i=1;i<=n;i++)cin>>c[i];
47     for(ll i=1;i<=m;i++){
48         q[i].id=i;
49         cin>>q[i].L>>q[i].R;
50     }
51     sort(q+1,q+1+m,cmp);
52     ll L=1,R=0;
53     ll sum=0;
54     for(ll i=1;i<=m;i++){
55         while(R<q[i].R){
56             R++;
57             if(num[c[R]]>1)sum-=com(num[c[R]],2);
58             num[c[R]]++;
59             if(num[c[R]]>1)sum+=com(num[c[R]],2);
60         }
61         while(R>q[i].R){
62             if(num[c[R]]>1)sum-=com(num[c[R]],2);
63             num[c[R]]--;
64             if(num[c[R]]>1)sum+=com(num[c[R]],2);
65             R--;
66         }
67         while(L<q[i].L){
68             if(num[c[L]]>1)sum-=com(num[c[L]],2);
69             num[c[L]]--;
70             if(num[c[L]]>1)sum+=com(num[c[L]],2);
71             L++;
72         }
73         while(L>q[i].L){
74             L--;
75             if(num[c[L]]>1)sum-=com(num[c[L]],2);
76             num[c[L]]++;
77             if(num[c[L]]>1)sum+=com(num[c[L]],2);
78         }
79         ll under=com(q[i].R-q[i].L+1,2);
80         ll gcdd=__gcd(sum,under);

```

```

81         q[i].res1=sum/gcdd;q[i].res2=under/gcdd;
82     }
83 }

```

2 时间戳 + 统计有多少个不同的数

```

1  const int SIZE=10500;
2  struct node{
3      int l,r,time,id;
4  }q[SIZE];
5  struct Update{
6      int x,y;
7  }upd[SIZE];
8  int l,r;
9  const int COLSIZE=1000500;
10 int col[COLSIZE];int unit;int bl[SIZE];int vis[COLSIZE];int ans=0;int res[SIZE];
11 void pop(int x){
12     ans -= !vis[x];
13 }
14 void push(int x){
15     ans += !vis[x]++;
16 }
17 bool cmp(node a,node b){// 奇偶排序
18     return
19         (bl[a.l]^bl[b.l])?bl[a.l]<bl[b.l]:((bl[a.r]^bl[b.r])?bl[a.r]<bl[b.r]:a.time<b.time);
20 }
21 void modify(int x){
22     if(l <= upd[x].x && upd[x].x <= r){
23         pop(col[upd[x].x]);
24         push(upd[x].y);
25     }
26     swap(upd[x].y,col[upd[x].x]);
27 }
28 int main(){
29     int n,m;
30     scanf("%d%d",&n,&m);
31     int tmpb=pow(double(n),double(2.0/3.0));
32     for(int i=1;i<=n;i++){
33         scanf("%d",&col[i]);
34         bl[i]=i/tmpb;
35     }
36     int time=0;char ch;int x,y;
37     int mm=1,updm=1;
38     for(int i=1;i<=m;i++){
39         cin>>ch>>x>>y;
40         if(ch=='Q'){// Q查询
41             q[mm].l=x;q[mm].r=y;q[mm].time=time;q[mm].id=mm;mm++;

```

```

41     }
42     else if(ch=='R'){// R更换画笔
43         upd[updm].x=x,upd[updm].y=y;time++;updm++;
44     }
45 }
46 sort(q+1,q+mm,cmp);
47 l=1,r=0;int tp=0;
48 for(int i=1;i<mm;i++){
49     while(l < q[i].l)pop(col[l++]);
50     while(l > q[i].l)push(col[--l]);
51     while(r < q[i].r)push(col[++r]);
52     while(r > q[i].r)pop(col[r--]);
53     while(tp < q[i].time)modify(++tp);
54     while(tp > q[i].time)modify(tp--);
55     res[q[i].id]=ans;
56 }
57 }

```

3 树状数组维护区间两数之差

```

1  /*
2      HDU - 6534 树状数组维护区间两数之差
3      i<j, |ai-aj|≤ K, n (1≤n≤27000), m (1≤m≤27000) and K (1≤K≤109)
4  */
5  //树状数组部分—————
6  int lowbit(int k){return k&-k;}
7  void add(int x,int k){
8      while(x<=n){
9          tree[x]+=k;
10         x+=lowbit(x);
11     }
12 }
13 int Get_sum(int x){
14     int ans=0;
15     while(x!=0){
16         ans+=tree[x];
17         x-=lowbit(x);
18     }
19     return ans;
20 }
21 //—————
22 /*
23 // 统计 a[i]^a[i+1]^...a[j]=k, 非树状数组
24 // 1 ≤ n, m ≤ 100 000, 0 ≤ k ≤ 1 000 000, (0 ≤ ai ≤ 1 000 000)
25 ll tmp=0;
26 void add(ll x){
27     tmp += cnt[sum[x]^k];

```

```

28     cnt[sum[x]]++;
29 }
30 void del(ll x){
31     cnt[sum[x]]--;
32     tmp -= cnt[sum[x]^k];
33 }
34 */
35 int sum=0;
36 void push(int x){
37     sum+=Get_sum(rr[x])-Get_sum(ll[x]-1);
38     add(a[x],1);
39 }
40 void pop(int x){
41     add(a[x],-1);
42     sum-=Get_sum(rr[x])-Get_sum(ll[x]-1);
43 }
44
45 int main(){
46     sort(q+1,q+1+m,cmp);
47     int L=1,R=0;
48     for(int i=1;i<=m;i++){
49         while(L > q[i].l)push(--L);
50         while(R < q[i].r)push(++R);
51         while(L < q[i].l)pop(L++);
52         while(R > q[i].r)pop(R--);
53         display[q[i].id]=sum;//用于输出
54     }
55 }

```

4 统计有多少个不同的数

```

1  /*
2     HYSBZ - 1878 统计[L,R]区间内有多少个不同的数
3  */
4  #define ll long long
5  const ll SIZE=1005000;
6  struct node{
7      ll L,R;
8      ll id,res;
9  }q[SIZE];
10 ll c[SIZE];ll n,m;ll num[SIZE];ll sum=0;ll dis[SIZE];ll pos[SIZE];
11 bool cmp(const node& a,const node& b){//奇偶排序
12     return (pos[a.L]^pos[b.L])?pos[a.L]<pos[b.L]:((pos[a.L] &1)?a.R<b.R:a.R>b.R);
13 }
14 ll ans=0;
15 int main(){
16     scanf("%lld",&n);//n个数

```

```

17  ll tmpb=sqrt(n);
18  sort(q+1,q+1+m,cmp);
19  ll L=1,R=0;
20  for(ll i=1;i<=m;i++){
21      while(L<q[i].L)ans-=!num[c[L++]];
22      while(L>q[i].L)ans+=!num[c[--L]]++;
23      while(R<q[i].R)ans+=!num[c[++R]]++;
24      while(R>q[i].R)ans-=!num[c[R--]];
25      dis[q[i].id]=ans;//用于输出
26  }
27
28  }

```