

太戈编程
etiger.vip

信奥算法

2657

数值分类 位置列表

`clr[i]`表示*i*号位置的颜色/数值

```
vector<ll> p[N];
```

`p[v]`储存数值为*v*的位置编号

```
24 for(ll i=1;i<=n;++i)  
25     p[clr[i]].push_back(i);
```

当`p[v]`位置单调
可以二分定位

```
23 void solve(){
24     for(ll i=1;i<=n;++i)
25         p[clr[i]].push_back(i);
26     for(ll i=1;i<=m;++i){
27         ll x,y,z;
28         cin>>x>>y>>z;
29         ll uby=
30         ll lbx=
31         ll ans=uby-lbx;
32         cout<<ans<<" ";
33     }
34     cout<<endl;
35 }
```

另解：离线询问

另解：莫队

另解：分块

多种算法比较

复杂度?

在线/离线?

2658

原颜色序列 ll clr[N];

数值分类 位置列表 vector<ll> p[N];

```
for(ll i=1;i<=n;++i)
    p[clr[i]].push_back(i);
```

如何静态统计色块数量?

计数技巧

每段色块选一个代表：每段最左侧

```
for(ll i=1;i<=n;++i)
    nC+=(clr[i]!=clr[i-1]);
```


clr[i]=

| i=1 | i=2 | i=3 | i=4 | i=5 | i=6 |
|-----|-----|-----|-----|-----|-----|
| 5 | 5 | 3 | 3 | 3 | 7 |

如何静态统计色块数量?

计数技巧

每段色块选一个代表：每段最左侧

```
for(11 i=1;i<=n;++i)
    nC+=(clr[i]!=clr[i-1]);
```

单点修改如何影响色块数量?

```

13 void solveBF(){
14     ll nC=0;
15     for(ll i=1;i<=n;++i)
16         nC+=(clr[i]!=clr[i-1]);
17     for(ll i=1;i<=m;++i){
18         ll op;
19         scanf("%lld",&op);
20         if(op==2){
21             printf("%lld\n",nC);
22             continue;
23         }
24         ll x,y;
25         scanf("%lld %lld",&x,&y);
26         
27         for(ll i=1;i<=n;++i){
28             if(clr[i]!=x) 
29             nC-=(clr[i-1]==y);
30             
31         }
32         for(ll i=1;i<=n;++i)
33             if(clr[i]==x) 
34     }
35 }

```

此暴力解法为什么慢?

每种颜色只在特定位置出现
不需要访问所有n个位置

加速方法:数值分类位置列表

加速后最差复杂度多少?
请构造最差情况

形成正解

| | | | | | | |
|---------|-----|-----|-----|-----|-----|-----|
| | i=1 | i=2 | i=3 | i=4 | i=5 | i=6 |
| clr[i]= | 5 | 5 | 3 | 3 | 3 | 7 |

随着颜色变化，色块数量单调减少

识别要素

合并

小并大

较小的位置序列合并到较大的位置序列

小并大

clr[i]=

| i=1 | i=2 | i=3 | i=4 | i=5 | i=6 |
|-----|-----|-----|-----|-----|-----|
| 5 | 5 | 3 | 3 | 3 | 7 |

变色操作

颜色3

都改成

颜色5

clr[i]=

| i=1 | i=2 | i=3 | i=4 | i=5 | i=6 |
|-----|-----|-----|-----|-----|-----|
| 5 | 5 | 5 | 5 | 5 | 7 |

clr[i]=

| i=1 | i=2 | i=3 | i=4 | i=5 | i=6 |
|-----|-----|-----|-----|-----|-----|
| 5 | 5 | 3 | 3 | 3 | 7 |

变色操作

颜色5

都改成

颜色3

clr[i]=

| i=1 | i=2 | i=3 | i=4 | i=5 | i=6 |
|-----|-----|-----|-----|-----|-----|
| 3 | 3 | 3 | 3 | 3 | 7 |

重大发现

两种改变方法颜色不同
但不影响色块数量

之后再找颜色5时
该如何定位?

小并大

clr[i]=

| i=1 | i=2 | i=3 | i=4 | i=5 | i=6 |
|-----|-----|-----|-----|-----|-----|
| 5 | 5 | 3 | 3 | 3 | 7 |

变色操作

颜色x

都改成

颜色y

若颜色x次数大于颜色y

调换

颜色y

都改成

颜色x


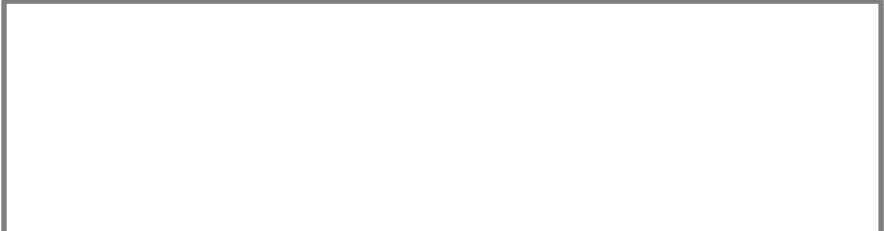
之后再找颜色y时
该如何定位?

id[y]记录原始颜色y
目前用哪个颜色代替

```
50 void solve(){
51     for(ll i=1;i<=n;++i)
52         p[clr[i]].push_back(i);
53     for(ll i=1;i<=n;++i)
54         nC+=(clr[i]!=clr[i-1]);
55     for(ll i=1;i<=n;++i)
56         id[clr[i]]=i;
57     for(ll i=1;i<=m;++i){
77 }
```

```
57 for(ll i=1;i<=m;++i){
58     ll op;
59     scanf("%lld",&op);
60     if(op==2){
61         printf("%lld\n",nC);
62         continue;
63     }
64     ll x,y;
65     scanf("%lld %lld",&x,&y);
66
67
68
69
70
71
72
73
74
75
76 }
```



```
37 void move(ll frm, ll to){
38     for(ll i=0; i<p[frm].size(); ++i){
39         
40
41     }
42     for(ll i=0; i<p[frm].size(); ++i){
43         
44
45     }
46     p[frm].clear();
47 }
48
49 }
```

两个循环能否合并成一个?

复杂度分析

每次合并：较小的列表大小至少翻倍

2659

如何动态统计色块数量?

计数技巧

每段色块选一个代表：每段最左侧

若 i 号位置颜色与 $i-1$ 号位置颜色不同
在 i 号打一个代表标记

修改+询问

维护每个位置是否有代表标记

区间内色块数 \approx 区间内有几个代表标记

使用
树状
数组
BIT

$[x, y]$ 区间色块数
= 区间内有几个代表标记
+ (x 号与 $x-1$ 号是否同色)

| $i=1$ | $i=2$ | $i=3$ | $i=4$ | $i=5$ | $i=6$ |
|-------|-------|-------|-------|-------|-------|
| 代表 | | 代表 | | 代表 | |

$[4, 5]$ 内有1个标记

```
56 for(ll i=1;i<=n;++i){  
57     p[clr[i]].push_back(i);  
58     id[clr[i]]=i;  
59     if(clr[i]!=clr[i-1]) add(i,1);  
60 }
```

```
61 for(ll i=1;i<=m;++i){
62     ll op,x,y;
63     scanf("%lld %lld %lld",&op,&x,&y);
64     if(op==2){
65         ll nC=
66         printf("%lld\n",nC);
67         continue;
68     }
69
70     if(p[id[x]].size()>p[id[y]].size()){
71
72
73
74     }
75     else{
76         move(id[x],id[y]);
77         id[x]=0;
78     }
79 }
```

```
38 void move(ll frm,ll to){
39     for(ll i=0;i<p[frm].size();++i){
40         ll pos=p[frm][i];
41         if(clr[pos-1]==to) add(pos,-1);
42         
43     }
44     for(ll i=0;i<p[frm].size();++i){
45         ll pos=p[frm][i];
46         
47     }
48     p[frm].clear();
49 }
50 }
```

输入文件包含多组测试数据

输入文件包含多组测试数据

易错点

公用变量清理

```
for(ll i=1;i<=n;++i){  
    p[clr[i]].clear();  
    bit[i]=0;  
}
```

对拍

生成多组随机数据

```
33 int main(){  
34     freopen("colour.in", "w", stdout);  
35     ll T=5;  
36     printf("%lld\n", T);  
37     while(T--){  
38         inputGen();  
39     }  
40     return 0;  
41 }
```



太戈编程

2657, 2658, 2659

要求

搭配暴力+对拍