

信奥算法



去重函数

`unique()`

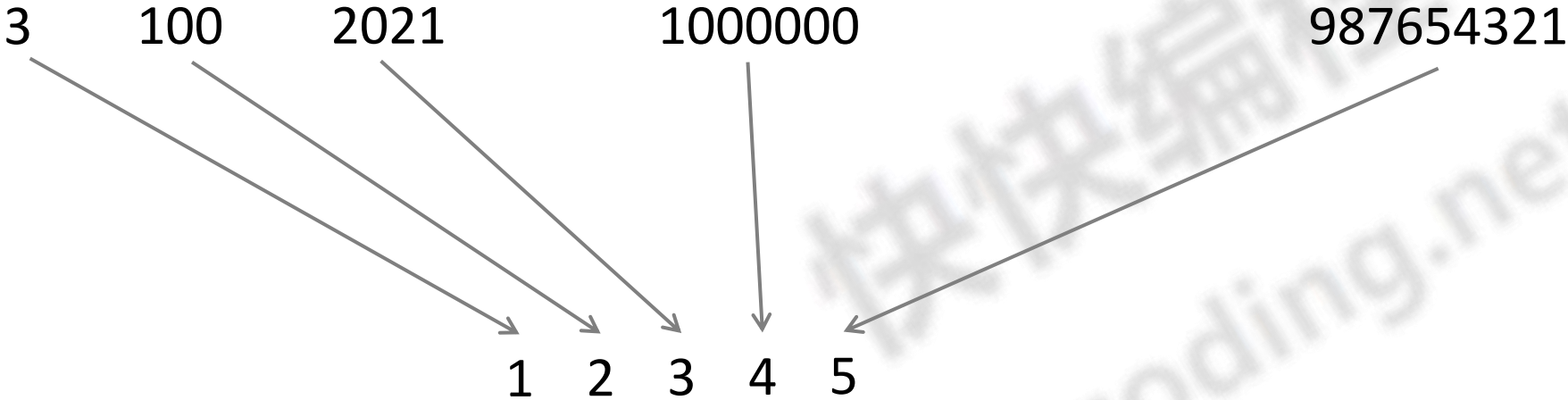
相邻的连续段数值若相同
只保留该段第一个数
删除该段其他数值



```
1 #include<iostream>
2 #include<algorithm>
3 using namespace std;
4 int main(){
5     const int N=8;
6     int x[N]={1,1,2,2,2,3,3,4};
7     int nU=unique(x,x+N)-x;
8     cout<<nU<<endl;
9     for(int i=0;i<N;i++)
10         cout<<x[i]<<" ";
11     cout<<endl;
12     for(int i=0;i<nU;i++)
13         cout<<x[i]<<" ";
14     cout<<endl;
15     return 0;
16 }
```



离散化
数值压缩





离散化方法1

idrk排序后确定排位

```
7 bool cmp(const int&a,const int&b){  
8     return x[a]<x[b];  
9 }  
  
15 for(int i=1;i<=n;i++) cin>>x[i];  
16 for(int i=1;i<=n;i++) id[i]=i;  
17 sort(id+1,id+1+n,cmp);  
18 for(int i=1;i<=n;i++) rk[id[i]]=i;
```

id[i]代表排序后第i个数原来是第几个

rk[i]代表原来第i个数排序后排第几个



离散化方法2

副本排序+去重+二分定位

```
12 for(int i=1;i<=n;i++)
13     cin>>x[i],y[i]=x[i];
14 sort(y+1,y+1+n);
15 int nU=unique(y+1,y+1+n)-y-1;
16 for(int i=1;i<=n;i++)
17     rk[i]=lower_bound(y+1,y+1+nU,x[i])-y;
```



易错点：去重前后元素个数不同

莫队算法

Mo's algorithm

莫队算法，主要是用分块思想+离线询问对区间排序来解决不带修改的区间问题

排序
规则

先比区间左端点所在块号
块内再比区间右端点

复杂度分析+最优分块大小

序列编号分组每组大小 L , 共 n/L 组

左端点

每两个询问间: 左端点抖动距离不超过 L
左端点抖动总距离不超过 mL

右端点

1号块内: 右端点移动距离不超过 n
2号块内: 右端点移动距离不超过 $n-L$

.....

右端点移动总距离不超过

$$n + (n - L) + (n - 2L) + \dots \approx \frac{n^2}{2L}$$

总复杂度

$$mL + \frac{n^2}{2L} \geq n\sqrt{2m} = O(n\sqrt{m})$$

额外排序
 $O(m \log m)$

等号条件

$$mL = \frac{n^2}{2L}, \quad L^2 = \frac{n^2}{2m}, \quad L = \frac{n}{\sqrt{2m}}$$

各种区间询问

给定数字数列 $x[]$ 长度为 n ，共 m 个询问，可能形式如下：

$[1, r]$ 编号范围内数值为 v 的共几个？

$[1, r]$ 编号范围内不同数值共几个？

$[1, r]$ 编号范围内是否所有数值都不相同？

$[1, r]$ 编号范围内恰好出现 k 次的数值共几个？



现场挑战 快快编程1834

快快编程
kkcoding.net



模型拆解

树的序列化(dfs序列)

子树询问转换为区间询问

离线区间询问使用莫队

排序后增量式做区间统计

快快编程
kkcoding.net



dfs序列中原节点编号

1

2

4

7

5

3

6

dfs序列中位置编号

1

2

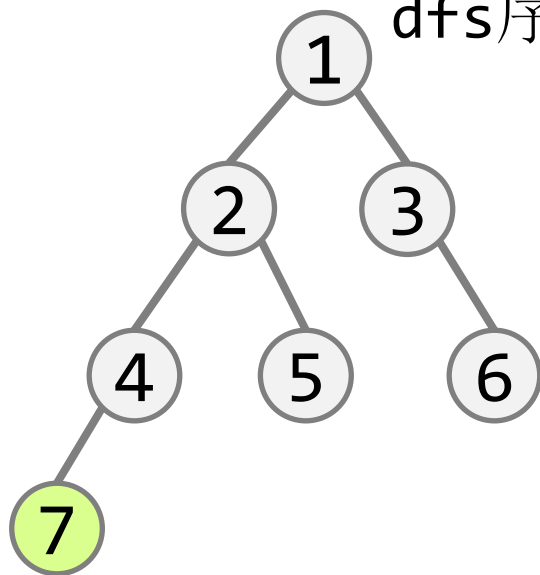
3

4

5

6

7



dfs序列里,子树节点总是连续段

1号为根的子树对应dfs序号[1,7]

2号为根的子树对应dfs序号[2,5]

3号为根的子树对应dfs序号[6,7]

4号为根的子树对应dfs序号[3,4]

5号为根的子树对应dfs序号[5,5]

6号为根的子树对应dfs序号[7,7]

7号为根的子树对应dfs序号[4,4]



dfs序列中原节点编号

1

2

4

7

5

3

6

dfs序号

1

2

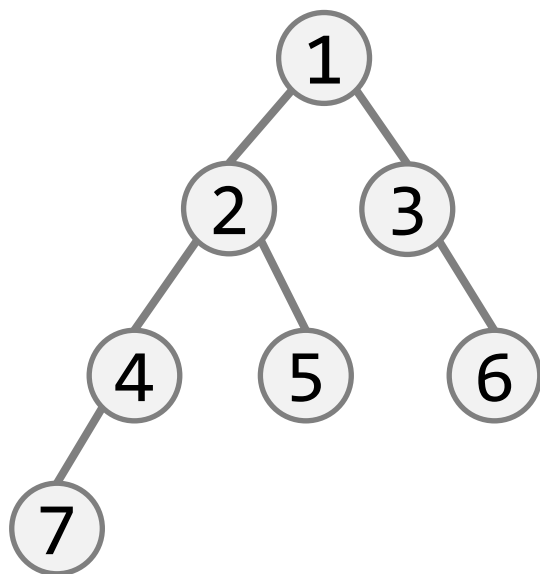
3

4

5

6

7



dfs序列里,子树节点总是连续段

u为根的子树对应dfs序号是个区间

[$dfn[u]$, $dfn[u]+sz[u]-1$]

[$tI[u]$, $tO[u]$]

$tI[u]$ 表示dfs访问u时的时间戳

$tO[u]$ 表示dfs访问u子树最后一个节点时的时间戳

```
8 void addedge(int u,int v){
9     to[++nE]=v;
10    nxt[nE]=hd[u];
11    hd[u]=nE;
12 }
13 void dfs(int u,int fa){
14     tI[u]=
15     id[timer]=u;
16     for(int i=hd[u];i;i=nxt[i]){
17         int v=to[i];
18         if(v==fa)continue;
19         dfs(v,u);
20     }
21     tO[u]=
22 }
```



离线区间询问的排序

快快编程
kkcoding.net



```
7 struct Query{int id,u,c,l,r,idB;} q[N];
```

```
32     cin>>m;
33     for(int i=1;i<=m;++i){
34         int u,c;
35         cin>>q[i].u>>q[i].c;
36         q[i].id=i;
37     }

46     dfs(1,0);
47     int B=n/sqrt(2*m);
48     for(int i=1;i<=m;++i){
49         int u=q[i].u;
50         q[i].l=tI[u];
51         q[i].r=tO[u];
52         q[i].idB=(q[i].l-1)/B+1;
53     }
54     sort(q+1,q+1+m,cmp);
```




请同学用电脑完成
离线区间询问的排序规则cmp

快快编程
kkcoding.net



```
39 bool cmp(const Query&a, const Query&b){  
40     if(a.idB < b.idB) return 1;  
41     if(a.idB > b.idB) return 0;  
42     if(a.idB & 1) return a.r < b.r;  
43     return a.r > b.r;  
44 }
```

快快编程
kkcoding.net

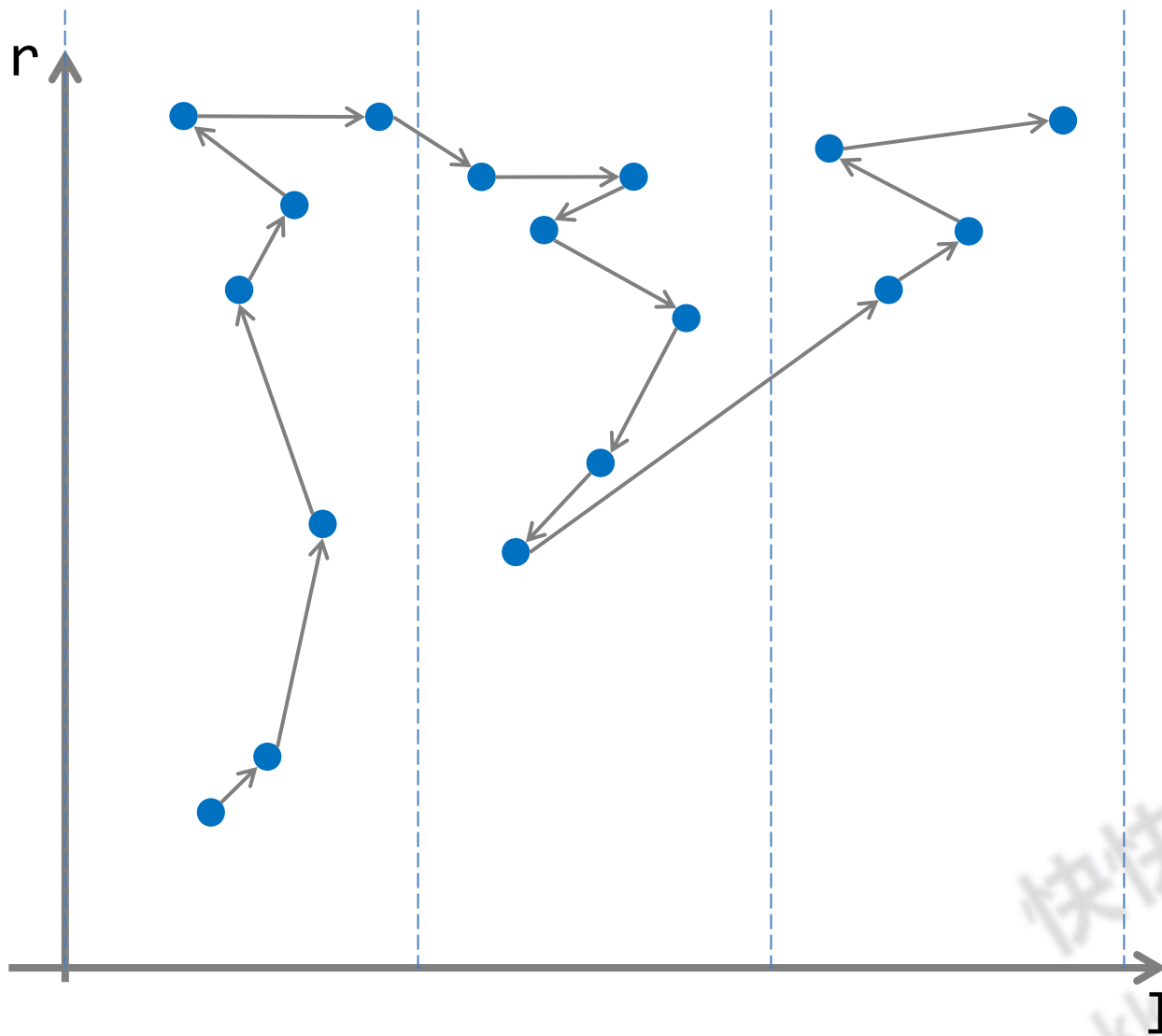
增量式更新

离线问询

区间排序

左端点
分块

先比左端点所在块号
块内再比右端点





增量式实现区间统计

快快编程
kkcoding.net



`id[i]`表示树的dfs序列里第*i*个访问的节点的原始编号

`x[i]`表示树的dfs序列里第*i*个访问的节点的颜色编号

```
55     for(int i=1;i<=n;++i) x[i]=clr[id[i]];

56     int l=0,r=0;
57     for(int i=1;i<=m;++i){
58         while(l>q[i].l) ++cnt[x[--l]];
59         while(l<q[i].l) --cnt[x[l++]];
60         while(r<q[i].r) ++cnt[x[++r]];
61         while(r>q[i].r) --cnt[x[r--]];
62         ans[q[i].id]=cnt[q[i].c];
63     }
```



现场挑战 快快编程1835

快快编程
kkcoding.net



模型拆解

树的序列化(dfs序列)

子树询问转换为区间询问

离线区间询问使用莫队

排序后增量式做区间统计

请写出如何统计不同颜色个数?

kkcoding.net



$id[i]$ 表示树的dfs序列里第 i 个访问的节点的原始编号

$x[i]$ 表示树的dfs序列里第 i 个访问的节点的颜色编号

```
52  sort(q+1,q+1+m,cmp);
53  for(int i=1;i<=n;++i) x[i]=clr[id[i]];
54  int l=0,r=0,cUnq=0;
55  for(int i=1;i<=m;++i){
56      while(l>q[i].l) cUnq+=(++cnt[x[--l]]==1);
57      
58      
59      
60      ans[q[i].id]=
61  }
```



```
44 dfs(1,0);
45 int B=
46 for(int i=1;i<=m;++i){
47     int u=q[i].u;
48     q[i].l=
49     q[i].r=
50
51 }
52 sort(q+1,q+1+m,cmp);
```



现场挑战 快快编程1837

快快编程
kkcoding.net



模型
拆解

树的序列化(dfs序列)

子树询问转换为区间询问

离线区间询问使用莫队

排序后增量式做区间统计

为什么需要离散化/数值压缩?

请写出如何统计
恰好出现 k 次的颜色个数?



颜色编号数值太大，需要数值压缩

```
23 for(int u=1;u<=n;++u){  
24     cin>>clr[u];  
25     w[u]=clr[u];  
26 }  
27 sort(w+1,w+1+n);  
28 int nU=  
29 for(int u=1;u<=n;++u)  
30     clr[u]=
```



`cnt[c]`表示当前区间内颜色`c`出现几次

`ccnt[k]`表示当前区间内出现恰好`k`次的颜色共几种

```
59 sort(q+1,q+1+m,cmp);
60 for(int i=1;i<=n;++i) x[i]=clr[id[i]];
61 int l=0,r=0,val=0;
62 for(int i=1;i<=m;++i){
63     while(l>q[i].l)
64         val=++cnt[x[--l]], ccnt[val]++, ccnt[val-1]--;
65
66
67
68
69
70
71     ans[q[i].id]=ccnt[k];
72 }
```

快快编程作业

1834

1835

1837