

题目描述

给定序列 a_i , m 次查询, 每次询问区间 $[l, r]$ 中不同数字的数量。

$1 \leq n, m, a_i \leq 10^6, 1 \leq l \leq r \leq n$ 。

离线树状数组

对于若干个询问区间 $[l, r]$, 如果它们的 r 都相等的话, 对于序列中的同一个数字, 一定只关心出现在最右面的那一个。

例如: 对于序列 $[1, 3, 4, 5, 1]$, 对于所有 $r = 5$ 的询问来说, 第一个位置上的 1 没有意义, 因为可以用第 5 位的 1 来替代计算。

所以我们可以对所有查询的区间按照 r 来排序, 然后维护一个树状数组 $c[i]$, $c[i]$ 表示位置 1 到 i 中出现过的不同数字。

例如: 对于序列 $[1, 2, 1, 3]$ 。

对于位置一的数字 1, 进行 $add(1, 1)$ 操作, 表示第一个位置出现了一个不一样的数字, 此时树状数组所对应的每个位置上的数字为: $[1, 0, 0, 0]$ 。

对于位置二的数字 2, 进行 $add(2, 1)$ 操作, 表示第二个位置出现了一个不一样的数字, 此时树状数组所对应的每个位置上的数字为: $[1, 1, 0, 0]$ 。

对于位置三的数字 1, 由于之前已经出现过数字 1 了, 首先把那个 1 所在的位置在树状数组中删除, 即 $add(1, -1)$, 然后把它加到位置三, 即 $add(3, 1)$ 。此时树状数组中的数字是 $[0, 1, 1, 0]$ 。

如果有一个询问 $[2, 3]$, 那么 $sum(3) - sum(2 - 1) = 2$ 就是答案。

代码

```
1  #include <iostream>
2  #include <algorithm>
3  #include <cstdio>
4  using namespace std;
5  struct BIT
6  {
7      int l,r;
8      int ID;//询问编号
9  }ask[1000010];
10 int n,a[1000010],c[1000010],ans[1000010];
11 int vis[1000010];
12 bool cmp(BIT A,BIT B)
13 {
14     return A.r<B.r;
15 }
16 int lowbit(int x)
17 {
18     return x&-x;
19 }
20 void add(int x,int val)
21 {
22     while(x<=n)
23     {
```

```

24         c[x]=c[x]+val;
25         x=x+lowbit(x);
26     }
27 }
28 int sum(int x)
29 {
30     int ans=0;
31     while(x)
32     {
33         ans=ans+c[x];
34         x=x-lowbit(x);
35     }
36     return ans;
37 }
38 int main()
39 {
40     cin>>n;
41     for(int i=1;i<=n;i++)
42         scanf("%d",&a[i]);
43     int m;
44     cin>>m;
45     for(int i=1;i<=m;i++)
46     {
47         scanf("%d %d",&ask[i].l,&ask[i].r);
48         ask[i].ID=i;
49     }
50     sort(ask+1,ask+1+m,cmp);
51     int Next=1;
52     for(int i=1;i<=m;i++)
53     {
54         for(int j=Next;j<=ask[i].r;j++)
55         {
56             if(vis[a[j]])//a[j]已经出现过
57                 add(vis[a[j]],-1);
58             add(j,1);
59             vis[a[j]]=j;
60         }
61         Next=ask[i].r+1;//更新下一次查询的位置
62         ans[ask[i].ID]=sum(ask[i].r)-sum(ask[i].l-1);
63     }
64     for(int i=1;i<=m;i++)
65         printf("%d\n",ans[i]);
66     return 0;
67 }

```