

【bzoj2821】作诗(Poetize)

2014年7月19日

1,288

6

Description

神犇SJY虐完HEOI之后给傻×LYD出了一题：

SHY是T国的公主，平时的一大爱好是作诗。

由于时间紧迫，SHY作完诗之后还要虐OI，于是SHY找来一篇长度为N的文章，阅读M次，每次只阅读其中连续的一段 $[l, r]$ ，从这一段中选出一些汉字构成诗。因为SHY喜欢对偶，所以SHY规定最后选出的每个汉字都必须在 $[l, r]$ 里出现了正偶数次。而且SHY认为选出的汉字的种类数（两个一样的汉字称为同一种）越多越好（为了拿到更多的素材！）。于是SHY请LYD安排选法。

LYD这种傻×当然不会了，于是向你请教.....

问题简述：N个数，M组询问，每次问 $[l, r]$ 中有多少个数出现正偶数次。

Input

输入第一行三个整数n、c以及m。表示文章字数、汉字的种类数、要选择M次。

第二行有n个整数，每个数 A_i 在 $[1, c]$ 间，代表一个编码为 A_i 的汉字。

接下来m行每行两个整数l和r，设上一个询问的答案为ans(第一个询问时ans=0)，令 $L=(l+ans)\bmod n+1$ ， $R=(r+ans)\bmod n+1$ ，若 $L>R$ ，交换L和R，则本次询问为 $[L, R]$ 。

Output

输出共m行，每行一个整数，第i个数表示SHY第i次能选出的汉字的最多种类数。

Sample Input

```
5 3 5
1 2 2 3 1
0 4
1 2
2 2
2 3
3 5
```

Sample Output

```
2
0
```

0
0
1

HINT

对于100%的数据， $1 \leq n, c, m \leq 10^5$

题解

分块大法好

类似区间众数的做法，预处理 $F[i][j]$ 表示第 i 块到第 j 块的答案

一个询问 $l-r$ ，那么中间大块 $x-y$ 的答案已经得到了

只要考虑 $l-x$ 和 $y-r$ 对答案的影响，对于这至多 $2/\sqrt{n}$ 个数，对于每个数统计它在 $x-y$ 出现次数 t_1 ，以及 $l-r$ 出现次数 t_2 ，根据 t_1, t_2 的奇偶性考虑其对答案的影响

每块大小 $\sqrt{n/\log n}$ ，复杂度 $n/\sqrt{n} \log n$

顺便附关于块大小分析

设分块大小为 x ，分块数 n/x ，预处理 $n/x \cdot n$

m 与 n 同级，视为 n 个询问，每次询问二分 x 次 $n/x \cdot \log n$ （除非相同的数字很多，否则 $\log n$ 会很小）

$n \cdot (x \cdot \log n + n/x)$

分块大小应该是 $\sqrt{n/\log n}$

```

1 #include<iostream>
2 #include<cstdio>
3 #include<cstdlib>
4 #include<algorithm>
5 #include<cstring>
6 #include<cmath>
7 #define inf 0x7fffffff
8 using namespace std;
9 inline int read()
10 {
11     int x=0,f=1;char ch=getchar();
12     while(ch>'9' || ch<'0'){if(ch=='-') f=-1;ch=getchar();}
13     while(ch>='0' && ch<='9'){x=x*10+ch-'0';ch=getchar();}
14     return x*f;
15 }
16 int n,c,m,block,cnt;
17 int L[1505],R[1505],f[1505][1505];
18 int a[100005],belong[100005],tmp[100005],first[100005],last[100005];
19 int lastans;
20 bool mark[100005];
21 struct data{int p,v;}b[100005];
22 inline bool operator<(data a,data b)
23 {
24
```

```
25     return (a.v<b.v) || (a.v==b.v&&a.p<b.p);
26 }
27 void pre()
28 {
29     int tot;
30     for(int i=1;i<=cnt;i++)
31     {
32         for(int j=L[i];j<=n;j++) tmp[a[j]]=0;
33         tot=0;
34         for(int j=L[i];j<=n;j++)
35         {
36             if(!(tmp[a[j]]&1)&&tmp[a[j]]) tot--;
37             tmp[a[j]]++;
38             if(!(tmp[a[j]]&1)) tot++;
39             f[i][belong[j]]=tot;
40         }
41     }
42     for(int i=1;i<=n;i++)
43         b[i].p=i,b[i].v=a[i];
44     sort(b+1,b+n+1);
45     for(int i=1;i<=n;i++)
46     {
47         if(!first[b[i].v]) first[b[i].v]=i;
48         last[b[i].v]=i;
49     }
50 }
51 int findup(int x,int v)
52 {
53     int l=first[v],r=last[v];
54     int tmp=0;
55     while(l<=r)
56     {
57         int mid=(l+r)>>1;
58         if(x<b[mid].p) r=mid-1;
59         else {l=mid+1;tmp=mid;}
60     }
61     return tmp;
62 }
63 int finddown(int x,int v)
64 {
65     int l=first[v],r=last[v];
66     int tmp=inf;
67     while(l<=r)
68     {
69         int mid=(l+r)>>1;
70         if(x>b[mid].p) l=mid+1;
71         else {tmp=mid;r=mid-1;}
72     }
73     return tmp;
74 }
75 int find(int x,int y,int v)
76 {
77     return max(findup(y,v)-finddown(x,v)+1,0);
78 }
79 int que(int x,int y)
80 {
81     int ans=0,t1,t2,a1,bx=belong[x],by=belong[y];
82     if(bx==by||bx+1==by)
83     {
84         for(int i=x;i<=y;i++)
85         {
```

```
88         a1=a[i];if (mark[a1]) continue;
89         int t=find(x,y,a1);
90         if(!(t&1)){mark[a1]=1;ans++;}
91     }
92     for(int i=x;i<=y;i++)mark[a[i]]=0;
93 }
94 else
95 {
96     int l=L[bx+1],r=R[by-1];
97     ans=f[bx+1][by-1];
98     for(int i=x;i<l;i++)
99     {
100         a1=a[i];if (mark[a1]) continue;
101         t1=find(x,y,a1),t2=find(l,r,a1);
102         if(!(t1&1))
103             {if((t2&1)||!t2)ans++;}
104         else if(!(t2&1)&& t2)ans--;
105         mark[a1]=1;
106     }
107     for(int i=r+1;i<=y;i++)
108     {
109         a1=a[i];if (mark[a1]) continue;
110         t1=find(x,y,a1),t2=find(l,r,a1);
111         if(!(t1&1))
112             {if((t2&1)||!t2)ans++;}
113         else if(!(t2&1)&& t2)ans--;
114         mark[a1]=1;
115     }
116     for(int i=x;i<l;i++)mark[a[i]]=0;
117     for(int i=r+1;i<=y;i++)mark[a[i]]=0;
118 }
119 return ans;
120 }
121 }
122 int main()
123 {
124     n=read();c=read();m=read();
125     for(int i=1;i<=n;i++)a[i]=read();
126     block=sqrt((double)n/log((double)n)*log(2));
127     if(n%block)cnt=n/block+1;
128     else cnt=n/block;
129     for(int i=1;i<=n;i++)belong[i]=(i-1)/block+1;
130     for(int i=1;i<=cnt;i++)
131         L[i]=(i-1)*block+1,R[i]=i*block;
132     R[cnt]=n;
133     pre();
134     for(int i=1;i<=m;i++)
135     {
136         int l=read(),r=read();
137         int x=(l+lastans)%n+1,y=(r+lastans)%n+1;
138         if(x>y)swap(x,y);
139         lastans=que(x,y);
140         printf("%d\n",lastans);
141     }
142     return 0;
143 }
```