

Rabbit_hair

If you want something you have never had,you must be willing to do something you have never done.

博客园 首页 新随笔 联系 订阅 管理

随笔- 54 文章- 0 评论- 3

粉丝: 11
关注: 3
+加关注

<2013年5月>

| 日 | 一 | 二 | 三 | 四 | 五 | 六 |
|----|----|----|----|----|----|----|
| 28 | 29 | 30 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | 31 | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |

搜索

找我看

谷歌搜索

常用链接

我的随笔

我的评论

我的参与

最新评论

我的标签

更多链接

随笔档案

2014年2月 (2)

2013年12月 (1)

2013年11月 (4)

2013年10月 (3)

2013年9月 (1)

2013年8月 (7)

2013年7月 (7)

2013年6月 (4)

2013年5月 (14)

2013年4月 (5)

2013年1月 (2)

2012年9月 (1)

2012年8月 (2)

2012年5月 (1)

阅读排行榜

1. 主席树: 动态 Kth(1910)

2. 数位DP(1671)

3. 莫比乌斯反演(855)

4. 主席树 静态区间第k大(684)

5. 后缀数组题目(542)

评论排行榜

1. 2013多校第三场(3)

推荐排行榜

1. 概率DP(1)

2. 主席树 静态区间第k大(1)

1 /*
2 主席树: 对于序列的每一个前缀建一棵以序列里的值为下标的线段树 (所以要先离散化),
3 记录该前缀序列里出现的值的次数;
4 记离散后的标记为1~n; (下面值直接用1~n代替);
5 对于区间[x,y]的第k大的值, 那么从root[x-1],root[y]开始,
6 t=root[y].l[mid]-root[x-1].l[mid], t表示区间[x,y]内值在[l,mid]的个数
7 先判断t是否大于k, 如果t大于k,那么说明在区间[x,y]内存在[l,mid]的数的个数大于k,
8 也就是第k大的值在[l,mid]中, 否则在[mid+1,r]中;
9
10 这样我们依次同时从root[x-1],root[y]往下走
11 当l==r时 第k大的值就是离散后标记为l的值;
12
13 如果每棵线段都建完整的化, n*(n<<2)肯定会MLE.
14 我们发现对于前缀[l,i]和前缀[l,i+1]的线段树, 如果b[i+1]<=mid (b[i+1]表示a[i+1]离散后的标记)
15 那么线段树i和线段树i+1的左边是完全相同的, 根本不需要在建, 只需要用指针指一下就好;
16 那么对于一棵新的线段树其实我们最多要建的节点数为log(n): nlog(n)的节点数还是可以忍受的;
17
18
19
20 */
21 #include<cstdio>
22 #include<cstring>
23 #include<iostream>
24 #include<algorithm>
25 #include<cmath>
26 #include<cstdlib>
27 #define w(i) T[(i)].w
28 #define ls(i) T[(i)].ls
29 #define rs(i) T[(i)].rs
30 using namespace std;
31 const int N=100000+10;
32 struct node{
33 int ls,rs,w;
34 node() {ls=rs=w=0;}
35 }T[N*20];
36 int a[N],b[N],p[N],root[N],sz;
37 int cmp(int i,int j){
38 return a[i]<a[j];
39 }
40 int n,m;
41 void ins(int &i,int l,int r,int x){
42 T[++sz]=T[i]; i=sz;
43 w(i)++;
44 if (l==r) return;
45 int m=(l+r)>>1;
46 if (x<=m) ins(ls(i),l,m,x);
47 else ins(rs(i),m+1,r,x);
48 }
49 int query(int i,int j,int l,int r,int k){
50 if (l==r) return l;
51 int t=w(ls(j))-w(ls(i));
52 int m=(l+r)>>1;
53 if (t>=k) return query(ls(i),ls(j),l,m,k);
54 else return query(rs(i),rs(j),m+1,r,k-t);
55 }
56 int main(){
57 int Cas;scanf("%d",&Cas);
58 while (Cas--){
59 root[0]=0;
60 scanf("%d%d",&n,&m);
61 for (int i=1;i<=n;i++){
62 scanf("%d",&a[i]);p[i]=i;
63 }
64 sort(p+1,p+1+n,cmp); //间接排序, p[i]表示第i小的值在a[]中的下标;
65 for (int i=1;i<=n;i++) b[p[i]]=i; //离散化
66 /*
67 for (int i=1;i<=n;i++) cout<<a[i]<<" ";cout<<endl;
68 for (int i=1;i<=n;i++) cout<<p[i]<<" ";cout<<endl;
69 for (int i=1;i<=n;i++) cout<<b[i]<<" ";cout<<endl;
70 */
71 sz=0;

www.cnblogs.com/R1emon/archive/2013/05/23/3094635.html

1/2

```
72     for (int i=1;i<=n;i++){
73         root[i]=root[i-1];
74         ins(root[i],1,n,b[i]);
75     }
76     for (int i=0;i<m;i++){
77         int x,y,k;scanf("%d%d%d",&x,&y,&k);
78         int t=query(root[x-1],root[y],1,n,k);
79         printf("%d\n",a[p[t]]);
80     }
81 }
82 return 0;
83 }
```

绿色通道:

好文要顶

关注我

收藏该文

与我联系

[Rabbit_hair](#)
[关注 - 3](#)
[粉丝 - 11](#)
[+加关注](#)

1

推荐

0

反对

(请您对文章做出评价)

« 上一篇: [zoi3349](#)
» 下一篇: [主席树：动态 Kth](#)

posted @ 2013-05-23 12:32 [Rabbit_hair](#) 阅读(684) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) 网站首页。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
融云，免费为你的App加入IM功能——让你的App“聊”起来！！

- 最新IT新闻:
- 松下推新20英寸4K全球最大Windows平板电脑
 - 创业 CEO：《什么才是最难的事？》
 - 传Uber获中国人寿2亿美元投资，或与本地化有关
 - 富士康印度拟投资35亿美元 贴身服务小米？
 - 雅虎试水原创剧集反响平淡 重心或转向数字杂志
- » 更多新闻...

- 最新知识库文章:
- 移动应用测试——打造完美应用的秘诀
 - 什么是对象，为什么要面向对象，怎么才能面向对象？
 - 驱动方法不能改变任何事情
 - 推行TDD的思考
 - 首席工程师揭秘：LinkedIn大数据后台是如何运作的
- » 更多知识库文章...