

原

CDQ分治【分治（真得头疼）

2017年12月13日 17:02:13

Coco_T_

阅读数 7528

更多

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/wu_tongtong/article/details/78785836

前言：
说实话我对于这种没有固定板子，变化多端的算法实在是非常头疼的
但是不学不行，这也是一种很重要的思维方式
所以趁着这几天的心情比较好（快要放大周），赶紧学一波~

鸣谢：[tham](#)，stdcall

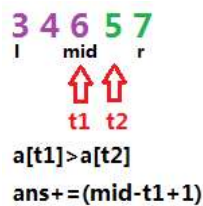
CDQ分治，传说中是一个♀神犇创造的算法
在了解这种算法之前，我们有必要了解一下一种基本的思想：分治

知识储备：分治

- 分治介绍
分而治之，将原问题不断划分成若干个子问题，直到子问题规模小到足以直接解决
子问题间互相独立且原问题形式相同，递归求解这些子问题，然后将各子问题的解合并得到原问题的解
- 一般步骤
 - 划分 Divide
将原问题划分成若干子问题，子问题间互相独立且与原问题形式相同
 - 解决 Conquer
递归解决子问题（递归是彰显分治优势的工具，仅仅进行一次分治策略也许看不出优势，但递归划分到子问题规模足够小，子问题的解可用岸
 - 合并 Merge
将各子问题的解合并得到原问题的解
- 时间复杂度
 - 直观估计
 - 分治由以上三部分构成，整体时间复杂度则由这三部分的时间复杂度之和构成
 - 由于递归，最终的子问题变得极为简单，以至于其时间复杂度在整个分治策略上的比重微乎其微

分治的一个经典例子就是归并求逆序对
简单叙述一下算法：
利用归并把序列对半分，在一般的归并过程中，我们在前后两部分各设两个指针，按照大小顺序合并成一个序列
我们要做的就是在这个过程中记录逆序对个数
什么情况下会有逆序对呢？
无非是在前面的数比在后面的数大（翻译过来：在前半部分的数比在后半部分的数大）
设前半部分的指针为t1，后半部分的指针为t2
如果a[t1] > a[t2]，那么t1~mid的元素一定都比t2大，一定都可以与t2形成逆序对

ans+=(mid-t1+1)+1



```

1  #include<cstdio>
2  #include<iostream>
3  #include<cstring>
4
5  using namespace std;
6
7  const int N=100010;
8  int a[N],b[N],n,ans=0;
9
10 void merge(int l,int r)
11 {
12     if (l==r) return;
13     int mid=(l+r)>>1;
14     merge(l,mid);
15     merge(mid+1,r);
16     int t1=l;
17     int t2=mid+1;
18     for (int i=l;i<=r;i++)
19     {
20         if ((t1<=mid&& a[t1]<=a[t2]) || t2>r) {
21             b[i]=a[t1];
22             t1++;
23         }
24         else
25         {
26             b[i]=a[t2];
27             ans+=(mid-t1+1);
28             t2++;
29         }
30     }
31 }
32 for (int i=l;i<=r;i++) a[i]=b[i];
33 }
34
35 int main()
36 {
37     scanf("%d",&n);
38     for (int i=1;i<=n;i++) scanf("%d",&a[i]);
39     merge(1,n);
40     printf("%d",ans);
41     return 0;
42 }

```



7



4



我为什么要介绍这个呢？

因为归并就是一个最简单的分治问题

我们在合并两个子区间的时候，要考虑到左边区间的对右边区间的影响

即，我们每次从右边区间的有序序列中取出一个元素的时候，要把“以这个元素结尾的逆序对的个数”（**左边区间有多少个元素比他大**）

这是一个典型的CDQ分治的过程

CDQ分治

CDQ分治是我们处理各类问题的重要武器

它的优势在于可以顶替复杂的高级数据结构，而且**常数比较小**；

缺点在于**必须离线操作**

二维偏序问题

上面介绍了归并求逆序对的经典问题，我们由此引入二维偏序问题：

给定N个有序对(a,b)，求对于每个(a,b)，满足 $a_0 < a$ 且 $b_0 < b$ 的有序对(a,b)有多少个

在归并求逆序对的时候，实际上每个元素是用一个有序对(a,b)表示的，

其中a表示数组中的位置，b表示该位置对应的值

我们求的就是“**对于每个有序对(a,b)，有多少个有序对(a0,b0)满足 $a_0 < a$ 且 $b_0 > b$** ”，这就是一个二维偏序问题

注意到在求逆序对的问题中，a元素是默认有序的，即我们拿到元素的时候，数组中的元素是默认从第一个到最后一个按顺序排列的，所以我们才能在求逆序对的时候忽略a元素带来的影响

因为我们在合并两个子问题的过程中，左边区间的元素一定出现在右边区间的元素之前，即左边区间的元素的a都小于右边区间元素的a



那么对于二维偏序问题，我们在拿到所有有序对(a,b)的时候，先把**a元素从小到大排序**
这时候问题就变成了“**求顺序对**”，因为a元素已经有序，可以忽略a元素带来的影响，和“求逆序对”的问题是一样的。

考虑二维偏序问题的另一种解法，用树状数组代替CDQ分治，即常用的用树状数组求顺序对
在按照a元素排序之后，我们对于整个序列从左到右扫描，每次扫描到一个有序对，求出“扫描过的有序对中有多少个有序对的b值小于当前b值”
然而当b的值非常大的时候，空间和时间上就会吃不消，便可以用CDQ分治代替，就是我们所说的“顶替高级数据结构”

二维偏序问题的拓展

给定一个N个元素的序列a，初始值全部为0，对这个序列进行以下两种操作

操作1：格式1 x k，把位置x的元素加上k

操作2：格式为2 x y，求出区间[x,y]内所有元素的和

这是一个经典的树状数组问题
但是我们就是要没事找事，我们用CDQ分治解决它——带修改和询问的问题

我们把ta转化成一个二维偏序问题，每个操作用一个有序对(a,b)表示，其中**a表示操作的时间，b表示操作的值**，时间是默认有序的，所以我们在合并过程中，就按照b从小到大的顺序合并。

首先我们把原数列和1操作都看作是**修改操作**
询问操作[l,r]我们拆成两个：l-1, r
因为我们询问的是一个区间和，一般的思路就是前缀和相减（我们需要具备这样的思维）
实际上我们这道题也可以这样，我们按照时间顺序进行修改
记录**前缀和**，当遇到-1的标记时，我们减去sum (l-1)
遇到r标记时，询问的处理就完成了

具体流程：

- 按照id（插入位置）归并排序
- 进行左区间的修改
- 统计右区间的询问

需要注意的是：

- 在合并的时候，我们只处理左区间的修改，只统计右区间的查询
因为**左区间的修改一定可以影响右区间的查询**
这就体现出了CDQ分治的基本思想了
- 我们把所有操作都记录到了一个数组中，所以数组的大小至少要开到**500000*3**

```
1  #include<cstdio>
2  #include<cstring>
3  #include<iostream>
4  #define ll long long
5
6  using namespace std;
7
8  const int N=5000010;
9  int n,m,totx=0,tot=0;    //totx是操作的个数,tot询问的编号
10
11 struct node{
12     int type,id;
13     ll val;
14     bool operator < (const node &a) const    //重载运算符,优先时间排序
15     {
16         if (id!=a.id) return id<a.id;
17         else return type<a.type;
18     }
19 };
20 node A[N],B[N];
21 ll ans[N];
22
23 void CDQ(int L,int R)
24 {
25     if (L==R) return;
26     int M=(L+R)>>1;
27     CDQ(L,M);
```

```

28    CDQ(M+1,R);
29    int t1=L,t2=M+1;
30    ll sum=0;
31    for (int i=L;i<=R;i++)
32    {
33        if ((t1<=M&&A[t1]<A[t2])||t2>R) //只修改左边区间内的修改值
34        {
35            if (A[t1].type==1) sum+=A[t1].val; //sum是修改的总值
36            B[i]=A[t1++];
37        }
38        else //只统计右边区间内的查询结果
39        {
40            if (A[t2].type==3) ans[A[t2].val]+=sum;
41            else if (A[t2].type==2) ans[A[t2].val]-=sum;
42            B[i]=A[t2++];
43        }
44    }
45    for (int i=L;i<=R;i++) A[i]=B[i];
46 }
47
48 int main()
49 {
50     scanf("%d%d",&n,&m);
51     for (int i=1;i<=n;i++)
52     {
53         tot++;
54         A[tot].type=1; A[tot].id=i; //修改操作
55         scanf("%lld",&A[tot].val);
56     }
57     for (int i=1;i<=m;i++)
58     {
59         int t;
60         scanf("%d",&t);
61         tot++;
62         A[tot].type=t;
63         if (t==1)
64             scanf("%d%lld",&A[tot].id,&A[tot].val);
65         else
66         {
67             int l,r;
68             scanf("%d%d",&l,&r);
69             totx++;
70             A[tot].val=totx; A[tot].id=l-1; //询问的前一个位置
71             tot++; A[tot].type=3; A[tot].val=totx; A[tot].id=r; //询问的后端点
72         }
73     }
74 }
75 CDQ(1,tot);
76 for (int i=1;i<=totx;i++) printf("%lld\n",ans[i]);
77 return 0;
}

```



三维偏序问题

给定N个有序三元组(a,b,c)，求对于每个三元组(a,b,c)，有多少个三元组(a0,b0,c0)满足a0 < a且b0 < b且c0 < c

不用CDQ的算法，我们就不说了（太麻烦了）

类似二维偏序问题，先按照a元素从小到大排序，这样我们就可以忽略a元素的影响

然后CDQ分治，按照b元素从小到大进行归并排序

哪c元素我们要怎么处理呢？

这时候比较好的方案就是借助权值树状数组，

每次从左边取出三元组 (a, b, c)，根据c值在树状数组中进行修改

从右边的序列中取出三元组 (a, b, c) 时，在树状数组中查询c值小于 (a, b, c) 的三元组的个数


注意，每次使用完树状数组要把树状数组清零



三维偏序问题的拓展

平面上有N个点，每个点的横纵坐标在[0,1e7]之间，有M个询问，每个询问为查询在指定矩形之内有多少个点，矩形用(x1,y1,x2,y2)的方式给出，其左下角坐标，(x2,y2)为右上角坐标

把每个点的位置变成一个修改操作，用三元组（时间，横坐标，纵坐标）来表示，把每个查询变成二维前缀和的查询
这样对于只有位于询问的左下角的修改，才对询问有影响
操作的时间是默认有序的，分治过程中按照横坐标从小到大排序，用树状数组维护纵坐标的信息



程序员学习编程的36个网站，超实用

电脑编程入门自学

835阅读

- 想对作者说点什么
- Zookkk: 感谢大佬，讲的很好（1年前 #2楼）[查看回复\(1\)](#)
- H_M_D: 膜拜ing，tql（9个月前 #1楼）[查看回复\(1\)](#)

学习笔记: cdq分治
今年xhr大神的论文有很大一部分内容是cdq分治及其扩展（也就是二进制分组），拜读后觉得还是蛮有用的，这里小...

[CDQ分治与整体二分]个人对CDQ分治与整体二分的理解
CDQ分治整体二分

点分治+CDQ分治+整体二分全纪录
点分治点分治讲解解决树上路径问题经典例题：点分治（长度小于m的路径计数）经典例题：点分治（聪聪可可）经...

分治、CDQ分治小结
分治、CDQ分治小结ASummaryforDivideandConquer0.Anouncement本文部分图片以及部分内容来自互联网，内...



免费永久云主机

CDQ分治——学习笔记
前言因为中国集训队的大佬们总会发明一些新的东西，所以中国的计算机竞赛水平才会一直处于国际前沿，而cdq分...

CDQ分治
CDQ分治就是先把第一维的信息排序好，确保左边对右边有影响，每次分治用第二维的信息再次判断让现在取到的左...

CDQ问题
二位偏序三维偏序不太懂，lowbit原来是树状数组里面的东西233二维偏序用排序加树状数组解决。原来是先对x排序...

整体二分\cdq分治——洛谷P3332 [ZJOI2013]K大数查询
https://daniu.luogu.org/problem/show?pid=3332第一次接触整体二分；上课的时候小红说这用树套树做，但感...

【总结】CDQ分治
总的来说，CDQ分治与普通分治不一样的地方在于，CDQ分治的对象是时间。即对于一个时间段[L,R]，我们取mid=...

点分治+CDQ分治+整体二分全纪录 - Coco_T的博客 - CSDN博客

分治、CDQ分治小结 - Tham 在思索中前行! - CSDN博客

陈小春坦言：这游戏不充钱都能当全服大哥，找到充值入口算我输！

贪玩游戏 · 顶新

CDQ分治 以树状数组模板为例
这两天状态极差网上cdqcdqcdq分治有关的东西也写得不太容易理解导致我对这个不是很复杂的东西学了很久内心m...

CDQ分治 - 咸鱼有梦想 - CSDN博客

CDQ分治 - O.K的博客 - CSDN博客

7

4

阅读数 8901

来自: [greatwall1995的...](#)

阅读数 1万+

博文 来自: [srand\(time\(0\)\)](#)

阅读数 498

博文 来自: [Coco_T的博客](#)

阅读数 6672

博文 来自: [Tham 在思索中前行...](#)

阅读数 1049

博文 来自: [CleverLarry的博客](#)

阅读数 543

博文 来自: [突然想要读文且曾...](#)

阅读数 40

博文 来自: [Clatterrr](#)

阅读数 17

博文 来自: [weixin_30673611...](#)

阅读数 3031

博文 来自: [BraketBN](#)

阅读数 211

来自: [lunch_的博客](#)