

选择题

小引理的简单使用,例如红黑树黑高和节点度的关系,不会很偏. 有五道,没想起来.

不偏不代表不难,这个中心思想贯穿以下全文: 😞

Q1

给定有序序列 $a_{1\dots n}, b_{1\dots n}$,设计算法在 $O(\log n)$ 时间内找到第 n 大元素. 即,subject to $k = \text{Merge}(a, b)[n]$

有递归和二分两种方法. 我只想到了二分. 假设 $a_{n/2} > b_{n/2}$,那么最大元素一定介于 $a_{n/2\dots n}$ 与 $b_{1\dots n/2}$ 之间,移动指针即可.

由于在两个数组上跑,需要稍微注意一点细节. 我是最后想出来的,没太管好像也没怎么扣分.

Q2

这位更是重量级

首先是一道作业题供大家欣赏:

给定序列 A ,在 $O(n \log n)$ 时间内计算逆序数 $\sigma(A)$

使用归并可以巧妙地解决这道题. 对我来说属于是有点有趣,仔细想也许能想出来的地步.

此题还有第二问:

count (i, j, k) , subject to

$$i < j < k \wedge a[i] < a[j] \wedge a[j] > a[k]$$

要求复杂度亦为 $O(N \log N)$.

作业没写出来,看了答案,大为震撼. 具体来说,你需要在归并的时候,统计所有 $a[i] < a[j]$ 以及 $a[j] > a[k]$ 的个数,然后将它们依 j 乘起来,即

$$count = \sum_j I(a[i] < a[j]) * I(a[j] > a[k])$$

I 表示符合这个关系式的个数,我们约定 $i < j < k$.

现在属于看答案能看懂的范畴. 让我们看看考题:

count (i, j, k), subject to

$$i < j < k \wedge a[i] < a[j] \wedge a[i] > a[k]$$

看仔细了,只有最后一个条件变了一点点,但是问题的性质变了亿点点.

作业题中的**信息**是完全分治的,即你一旦选定 j ,所有 i, k 是不是正好就全部被隔开? 可是考题变了,现在你不管选定哪个,都无法完全分治,则简单归并彻底失效.

其解决方法很复杂,简单来说,我们统计 $I_{i < j}, I_{i > k}, I_{k < j}, I_{k > j} \dots$,总而言之,统计所有的序对. 序对分为顺序对和逆序对,然后通过线性组合顺序对和逆序对的方式来表示出上面这个数(被称为领奖台数,因为正好中间最高,左边比右边高).

此题值得一做. 它还能进一步推广至多元序对以及任意序数的统计效率. 是否都是 $O(n \log n)$? 我猜测是的,因为可以在有限时间内用二元序对线性组合出任何序,而二元序对可以通过归并求出.

此题还说明了 $O(N \log N)$ 统计数据能力的极限在哪里,对于理解算法大有裨益.

Q3~Q5

好像都是普通的DP或者贪心. Q5遗憾写错,因本人先入为主想去套作业的贪心思路: 先变成Decision Problem,然后去贪这个判定问题的Boundary.

这是一个非常有用的方法,顺带一提. 在处理很麻烦的问题时(比如某些NPC的特例,可用P算法解决)很有用. 然而考试一般都是 $O(N)$ 的贪心, 变成判定问题往往更麻烦了,因为判定问题自己起码是个 $O(N)$.

一般数组上贪心三条路:

1. 顺序贪.
2. 变成前缀和,再顺序贪. 子数组和问题. 这题比顾乃杰讲得好太多, 我之前上gnj的课就没弄明白过.
3. 倒序贪. LCS, schedule问题(这个算是倒着的顺序贪)

然后就是不要死记硬背那个最优性原理. 陈雪老师好就好在不纠结这些无聊的细节. 有时候过于关注这个会导致妨碍对于问题的本质考量.

最后,前半学期的算法就是,OBS,lemma,key OBS, powerful lemma, 再加上一点trick和magic, Bingo,你做出来的.

学会了吗?