SEAPERM 题解

by 吉如一

October 28, 2015

1 题目大意

Sereja 有一个含 N 个整数的数组 A'。现在 Sereja 准备重新排列数组的元素，他想用一个置换 p 来得到一个新的数组 A，使得 $A_i = A'_{p_i}$。

令函数 $f(A,i) = S - \sum_{k=i}^{j} A_k$，其中 j 是使得 $\sum_{k=i}^{j} A_k \leq S$ 的最大整数，若 $A_i > S$，则 $f(A,i) = S$。请帮 Sereja 找到一个置换 p，使得 $\frac{\sum_{i=1}^{K} f(A,i)}{K}$ 的值尽可能的小。

数据范围 $1 \leq T \leq 10, 1 \leq K \leq N \leq 2000, 1 \leq A'[i] \leq 10000, 1 \leq S \leq 10^9$

2 评分方式

对于每一个点每一个数据，你的得分将按照如下方式计算：

如果你的输出不是一个合法的 1 到 n 的排列，那么这整个测试点得 0 分；

如果你的输出合法，假设由你的输出得到的数组为 A，你这个数据的得分是 $\frac{F}{\sum_{i=1}^{K} f(A,i)}$，其中 F 是对每一个数据分别设置的参数，以我的程序可以获得 100 分为基准。

对于每一个点，如果你的输出都合法，那么这一个测试点的得分将等于每一个数据得分的平均值。

3 对特殊数据的做法

这是一个 NP-hard 的问题，因此，对这类问题取得高分的做法通常有这两种：

1. 添加或者修改一些条件，这些条件可能会使得问题的最优解变劣，但是可以把问题转化为一个能在多项式时间内求出确定解问题。

2. 使用随机化来辅助程序，使得程序能够找到一些比较优秀的解。

而在此之前，可以先解决一些特殊的数据：

1. 当 $N \leq 8$ 时，可以暴力枚举所有可能的排列，然后进行取最优解，因为计算一次代价的时间复杂度是 $O(n)$，所以这样的复杂度是 $O(n!n)$。

2. 当 $S \geq \sum_{i=1}^{n} A'_i$ 时，这是一个经典的贪心问题，当序列是单调不降的时候一定最优，这时的时间复杂度是 $O(n \log n)$。

4 最简单的随机化做法

考虑随机 S 个排列，然后分别计算最优值，取最优的一个输出。

当然直接使用随机化的效果并不是很好，于是可以考虑先选取一个比较优秀的初始解，然后每一次对这个初始解进行调整，比如反转一个区间，或者交换两个位置之类的。这样就可以保证每一次随机出来的排列都是比较优的排列。

至于初始解的选择，受第二类特殊数据的启发，我们可以把数组 A' 进行从小到大的排序，并用那个排列进行初始解。实际上这个初始解是比较优秀的，在 CC 比赛的时候直接输出这个排列能获得 0.985pts 的分数（不过有一部分的原因是 CC 的计分方式不是很科学）。相比较之下，之后的随机调整对解的优化作用并不明显。

为了提高随机化的效率，可以采用一些高端一点随机化策略，比如模拟退火之类的算法，在经过参数调整之后，模拟退火在 CC 比赛时可以拿到 0.997pts 的分数。

5 贪心做法

随机化做法主要依靠初始解，而可以发现使用的初始解是由贪心得到的。可见这道题的贪心策略的用处是比较大的。

考虑如下的贪心，如果我们已经得到了第 $K+1$ 个数到第 n 个数是哪些，那么如何贪心得到一个比较优秀的第 1 个数到第 K 个数的选取方法呢？一个比较显然的思路就是依次最小化 $f(A,K)$，在这个基础上最小化 $f(A,K-1)$，以此类推。

具体的实现可以维护一个右指针 r，最开始 $r=n$，假设当前需要确定第 i 个数，那么第 $i+1$ 到第 n 个数都已经被确定了，我们找到一个最大的还没有被选取的数 w，使得 $w+\sum_{j=i+1}^{r} A_j \le S$，如果存在这样的 w，那么就令 $A_i=w$，否则就把 r 左移一位。如果已经有 $r=i$ 了，那么说明当前剩下的所有数都大于 S，随便选一个都是等价的。

这样实现一次的复杂度是 $O(n^2)$ 的，考虑再贪心得到一个比较优的第 $K+1$ 个数到第 n 个数排列。按照贪心的策略，可以尽可能地把小的数放在最右边，于是就从最小的数开始，能放就放，直到它们的和大于 S 位置。然后把剩下的数拿来运行上面的贪心算法，再把最后留下的数放到最右端，因为此时最右边还没有填的数对答案是没有影响的。

这个做法再配合随机的调整，在比赛的时候可以获得 0.999pts，可见效果还是很好的。

可以发现上面贪心做法的短板在于每次 $O(n^2)$ 的时间复杂度，我们可以使用平衡树把这个贪心的过程优化到每次 $O(n\log n)$，这样就可以对很多种可能的 $K+1$ 到 n 的选取方式进行贪心运算，而不必受时间的拘束不得不贪心得到一个右侧的选数方案。

目前这题我的程序的策略就是这样：每次随机一个第 $K+1$ 个数到第 n 个数的选取方案，如果使用以上的贪心算法进行进行计算，取最优值。这个流程时间范围内对每组数据可以进行 300 次左右，这样得到的解已经比赛场上第四名的 0.999pts 程序优秀许多了。

6 其他做法

而以上的做法虽然已经比较优了，但是还是比比赛时的最高分代码劣。最高分的策略是：

重复提交很多次，通过返回值例如运行状态，运行时间，使用内存之类的方法得到数据的特点。

这样的好处在于，普通做题时因为对数据一无所知，所以只能使用一个通用的策略，而这个策略可能对某一些特殊点很优秀，但是对另一些测试点很劣，这样就会导致平均得分并不高。而在 CC 上我们在写出很多种不同的策略之后，就可以使用这种方法得到对于哪一个点，使用哪一个策略最优，这样的得分就会比普通的做法高了。把这样的代码提交到清澄上之后得分就非常低了。

值的注意的是在 CC 上，我试着重新提交了 practice 中的 1 分代码，但是返回的得分只有 0.166 分。我觉得可能的原因之一是 CC 在中途更换过了数据但是没有对之前的人进行重测，所以评价一个代码的优劣还是以清澄上数据的返回值为准吧。