NOIP2022 黑色高级模拟赛 解题报告

by KaisuoShutong & wsyhb

目录

1 哼串计数（heng）

各显神通的签到题。

1.1 算法 1

$L=6$ 时，暴力枚举并判断给定串是否为哼哼串即可。

时间复杂度 $O(L^2)$，可以通过测试点 1，期望得分 5。

1.2 算法 2

暴力枚举原串所有长度为 6 的子串，再判断是否为哼哼串。

时间复杂度 $O\left(\binom{L}{6}\right)$，可能会带大约 6^2 的常数，但实现得好的话远远跑不满。

可以通过测试点 $1\sim4$，期望得分 20。

1.3 算法 3

分别枚举两个位置，将原串从该位置断开。在断开的三段中分别统计 hg、hg、au 。利用一些容斥技巧，可以求得答案。

时间复杂度 $O(L^2)$，可以通过测试点 $1\sim8$，期望得分 40。

1.4 算法 4

特殊性质 A 中，字符集很小。据此，可以分别枚举 h、g、a、u 对应的字符。此时问题转化为求解一个固定的串在原串中作为子序列的数量。设 $f_{i,j}$ 表示前 i 个原串字符共匹配了 j 个枚举串字符，可以简单 dp 得到。时间复杂度 $O(\alpha^5 n)$，其中 $\alpha=6$。

在算法 2/3 的基础上，可以额外通过测试点 $9\sim11$，期望得分 15。

1.5 算法 5

特殊性质 B 中，保证了相同字符一定相邻。容易发现这样一定无法构成 hghg 的形式，故答案为 0。

在算法 2/3 的基础上，可以额外通过测试点 16，期望得分 5。

1.6 算法 6

au 的处理始终是平凡的，考虑如何计数 hghg。

受到算法 4 的启发，设计 $f_{i,j,k,l}$ 表示前 i 个字符，匹配串 $h=j$，$g=k$，已匹配了 l 位的方案数。预处理出 f 后，枚举最后一个 g 位置，容斥式地计算带上 au 的方案即可。

时间复杂度 $O(\alpha^2 n)$，其中 $\alpha=62$。可能带有一定常数。

可以通过测试点 $1\sim15$，期望得分 75。

1.7 算法 7

考虑优化算法 6 中 dp。对于一次更新，容易发现 j 和 k 中一定有一个和当前字符 S_i 相同。仔细思考，这与普通子序列 dp 的优化思路不谋而合。因此，类似普通子序列 dp，可以将状态设计去掉一维。

时间复杂度 $O(\alpha n)$，可能带有一定常数。

可以通过所有测试点，期望得分 100。

2 金银变换（yinrier）

细心观察的套路题。

2.1 算法 0

当 $n \neq m$ 时，显然无解。

期望得分 0 分，但是如果没判的话，期望得分也是 0 分。

2.2 算法 1

我会读题！

对于 $k=1$，容易发现操作即为交换相邻的两个数，排序后判断对应位置是否相等即可。

对于 $2k \geqslant n$：

- 若 $2k > n$，则无法进行操作，直接判断 A 和 B 是否相等即可。
- 若 $2k = n$，容易发现操作即为交换序列的前后两半，判断交换和不交换两种情况即可。

可以通过测试点 9 和 12，期望得分 8 分。

2.3 算法 2

我会爆搜！

对于 $T \leqslant 100$ 且 $n, m \leqslant 10$，直接记忆化搜索即可。

状态数上界为 $5! \times 5! = 14400$，但不会分析应该也能猜到爆搜可过吧。

可以通过测试点 $1 \sim 3$，结合算法 1，期望得分 20 分。

2.4 算法 3

我会贪心和模拟！

可以证明，当 A 中元素互不相同时，下列贪心是正确的。（该结论的证明见 2.5 节）

按 $i = 1, 2, \cdots, n - 2k + 1$ 的顺序依次执行下列操作：

> 若 $A_i, A_{i+k}, A_{i+2k}, \cdots$ 中没有和 B_i 相等的数，无解。
> 否则，设 $A_{i+tk} = B_i$，考虑在不改变 $A_1, A_2, \cdots, A_{i-1}$ 的前提下，通过 t 次操作将 $A_{i+\mathrm{tk}}$ 移至 A_i。设当前下标为 j（$j > i$），若 $j + k - 1 \leqslant n$ 则执行操作 $i = j - k$（此处 i 是题目描述中的变量，下同），否则执行操作 $i = n - 2k + 1$。每次执行后 $j \rightarrow j - k$，因此 t 次操作后可以达成目标。

最终，若 $A = B$ 则答案为 `YES`，否则答案为 `NO`。

直接模拟上述过程，时间复杂度 $O(Tn^2)$。

可以通过测试点 $4 \sim 8$，期望得分 20 分。

注：放这档分主要是想提醒选手，如果不会正解的话有些贪心该写还是要写的。（即使你不会证）

2.5 算法 4

首先，容易发现一个元素的下标 $\bmod k$ 是不会变的，所以按下标 $\bmod k$ 把 A 和 B 的元素分别分成 k 组，对应组的可重集必须相等，否则无解。下面假设该条件满足。

考虑 A 中元素互不相同的情况。

考虑将 A 中的元素替换为其目标位置的下标。换句话说，若 $A_i = B_j$，则将 A_i 替换为 j。那么目标转化为使 A 单调递增，即 $A_i = i$。

考查每次操作前后的不变量：对于每一组元素所组成的子序列（按下标 $\bmod k$ 分组，下同），逆序对数量恰好变化 1。这意味着对于任意两组，它们逆序对数量的奇偶性要么保持相同，要么保持不同。

由于 $A_i = i$ 时每一组逆序对数量均为 0，因此初始时各组逆序对数量的奇偶性必须相同。

事实上，该条件是充分的。

证明. 按照 2.4 中的贪心方法，我们总可以使 $\forall i \in [1, n-2k+1] \cup \{n-k+1\}$，$A_i = i$。此时下标 $n-k+1$ 所在组的逆序对数量为 0，故其余组的逆序对数量为偶数。又它们的逆序对个数 $\leqslant 1$（每组只有末尾两个元素可能产生逆序对），进而逆序对数量均等于 0，得证。 □

进而证明了 2.4 中贪心做法的正确性。

时间复杂度 $O\left(\sum n \log n\right)$，可以通过测试点 4～8 及 15～20，期望得分 44 分。

2.6 算法 5

算法 4 和正解仅有一步之遥：A 中同一组有相同元素怎么办？

首先仍然判掉对应组可重集不相等的情况。

然后我们仍然可以检查逆序对奇偶性，如果全部相同则答案为 `YES`。

唯一的不同是，不全部相同也可能是 `YES`：

设 $a \equiv b \pmod k$（$a \neq b$），且 $A_a = A_b$，则存在 $c \equiv d \equiv a \pmod k$（$c \neq d$）使得 $B_c = B_d = A_a$。由于 A_a 和 A_b 的目标位置可交换，因此将 A_a, A_b 依次替换为 c, d，或者将 A_a, A_b 依次替换为 d, c 均可。

这意味着，对于有相同元素的一组，其逆序对数量的奇偶性可以人为更改，因此只需要对不存在相同元素的组，判断它们的逆序对数量的奇偶性是否全部相同即可。

时间复杂度仍为 $O\left(\sum n \log n\right)$。

可以通过全部测试点，期望得分 100 分。

3 枝江往事（zjiang）

卡不住暴力的萌萌题。

3.1 算法 1

暴力枚举 $n!$ 种顺序，并按照题给方式模拟求值。

时间复杂度 $O(n! \times n)$，可以通过测试点 $1 \sim 2$，期望得分 10。

3.2 算法 2

容易发现只有最后一次赋值操作及其后的乘法操作是有效的，且乘法操作与顺序无关。

当 $m \leqslant 12$ 时，暴力枚举最后一次赋值操作，以及有哪些乘法操作位于其后。

时间复杂度 $O(n2^m)$，可以通过测试点 $1 \sim 4$，期望得分 20。

3.3 算法 3

处理乘法是非常劣的。观察到题给操作全部在模素数意义下进行，可以想到利用原根 g，将所有数表示成 $y = g^x$ 的形式。$y \to x$ 的过程就实现了由乘法到加法的转化。

转化为加法以后，算法 2 中的暴力枚举就可以转化为一个背包。

时间复杂度 $O(mp)$，可以通过测试点 $1 \sim 6$，期望得分 30。

3.4 算法 4

算法 3 中的背包仅仅是一个可行性背包。这显然可以利用 bitset 来加速。

时间复杂度 $O\left(\frac{mp}{w}\right)$，可以通过测试点 $1 \sim 10$，期望得分 50。

可以进行一些适当的优化，实际得分 $50 \sim 100$。

3.5 算法 5

思考这样一个背包的实质：不过是对于每个权值 v，更新所有的 $i \to (i+v) \bmod p$。其中要求下标为 i 的位置的值为 1，下标为 $(i+v) \bmod p$ 的位置的值为 0。

可如是的更新（或者叫有效更新）始终只有 $O(p)$ 次。这样会非常浪费。

尝试对于每个修改，分治出这样的更新。用一个类似线段树的结构，以及一个额外的哈希+BIT，维护出区间的字符串哈希值。断环为链后，如果区间 $[l, r]$ 与 $[l+v, r+v]$ 不相同，则递归下去修改。

这样分析势能存在一个问题。对于 i 为 1，但 $(i+v) \bmod p$ 为 0 的情况，上述算法并不会更新，但却会递归下去。也就是说势能被利用，但未减少。

考虑本题特殊的性质：背包是在模意义下进行的。在模意义下，若将 i 与 $(i+v) \bmod p$ 连边，则整张图必定会形成若干个环。环上所有的 $(1, 0)$ 都会对应一个 $(0, 1)$，因此势能是正确的。

时间复杂度 $O(n \log^2 n)$，姑且认为 n、p 同阶。可以通过全部测试点，期望得分 100。

3.6 bonus

若将两种操作割裂开来，则乘法操作实则求的是一个数组的模意义下子集和。这个问题在一篇论文中给出过 $O(n \log n)$ 的解法。求得子集和后，再将两个操作对应的数组卷起来，也可以得到正确答案。

尚未可知能否不用卷积，转而将赋值操作作为初始值以得到相同结果。

4 膜皮圣经（pmyl）

防不住 AK 的防 AK 题。

4.1 算法 0

对于每个询问，枚举每个区间，找到生计平衡值最小的人家，爆搜一番，尝试拟合最短路。

时间复杂度未知，可能可以通过测试点 1，期望得分 0～5。

但都给了简要题意了，应该不会有人写这个了吧……所以删掉了这档。

4.2 算法 1

对于每个询问，枚举左端点 l，不断移动右端点 r 并动态更新最小的生计平衡值。配合前缀和等，可以做到 $O(n^2)$ 单次查询。时间复杂度即为 $O(qn^2)$。

可以通过测试点 1～2，期望得分 10。

4.3 算法 2

区间的贡献和询问无关，尝试对于每个区间预处理出答案和二维前缀和，即可快速询问。

时间复杂度 $O(n^2+q)$，可以通过测试点 1～4，期望得分 20。

4.4 算法 3

当最小值固定时，两侧的区间一定越长越好。因此，枚举最小值，提前用单调栈预处理好其管辖的范围，即可 $O(1)$ 计算最大权值。

时间复杂度 $O(qn)$，可以通过测试点 1～2 和 5～6，期望得分 20。

4.5 算法 4

当 $c_i=i$ 时，可以发现最小生计平衡值总在区间的左端点取到。式子可以被改写为 $\left(\sum_{i=l}^{r-1} d_i\right)\times c_l$。这样我们一定会固定 $r=R$，所以若令 $s_i=\sum_{j=1}^{i-1} d_j$，则一次询问即求 $\max((s_r-s_i)c_i)$，其中 $i\in[L,R]$。

离线后使用扫描线和李超树维护，时间复杂度 $O(q\log n)$，可以通过测试点 5～12，期望得分 40。

4.6 算法 5

考虑拼接算法 3 和算法 4。

由算法 3 给到的提示，可以想到笛卡尔树。

发现将一个区间定位到笛卡尔树上后，其对应的查询即为一个跨过当前最小值的大区间，和以最小值作为左/右端点的一个后缀和一个前缀。

考虑维护这样的东西。由算法 4 给到的提示，可以想到李超树。

首先，开两颗支持区改的李超树，其代表的意义为：在当前区间为 $[l,r]$ 的情况下，$i\in[l,r]$ 到 l 的前缀的所有区间的答案，和到 r 的后缀的所有区间的答案。

由于其父区间的最小值即为 $l-1$ 或 $r+1$，不难发现这样的维护一定可以达到目标。

时间复杂度 $O((n+q)\log^2 n)$，可以通过所有测试点，且因为信息较好维护所以不是树套树。