难题讨论《萨菲克斯・阿瑞》

cz_xuyixuan

NOI 多校联合训练

July 10, 2020

「CTSC2016」萨菲克斯・阿瑞

求出所有由 m 种字符组成，其中第 i 种字符出现次数不超过 c_i，且长度为 n 的字符串，共有多少种不同的后缀数组。答案对 10^9+7 取模。

「CTSC2016」萨菲克斯・阿瑞

求出所有由 m 种字符组成，其中第 i 种字符出现次数不超过 c_i，且长度为 n 的字符串，共有多少种不同的后缀数组。答案对 10^9+7 取模。

$0 \le c_1, c_2, \ldots, c_m \le n$，$c_1 + c_2 + \ldots + c_m \ge n$

$1 \le n, m \le 500$

「CTSC2016」萨菲克斯・阿瑞

求出所有由 m 种字符组成，其中第 i 种字符出现次数不超过 c_i，且长度为 n 的字符串，共有多少种不同的后缀数组。答案对 10^9+7 取模。

$0 \le c_1, c_2, \ldots, c_m \le n$， $c_1 + c_2 + \ldots + c_m \ge n$

$1 \le n, m \le 500$

(1) 、$n \le 10$

「CTSC2016」萨菲克斯・阿瑞

求出所有由 m 种字符组成，其中第 i 种字符出现次数不超过 c_i，且长度为 n 的字符串，共有多少种不同的后缀数组。答案对 10^9+7 取模。

$0 \le c_1, c_2, \ldots, c_m \le n$， $c_1 + c_2 + \ldots + c_m \ge n$

$1 \le n, m \le 500$

(1) 、$n \le 10$

(2) 、$c_1 + c_2 + \ldots + c_m = n$

「CTSC2016」萨菲克斯·阿瑞

求出所有由 m 种字符组成，其中第 i 种字符出现次数不超过 c_i，且长度为 n 的字符串，共有多少种不同的后缀数组。答案对 10^9+7 取模。

$0 \le c_1, c_2, \ldots, c_m \le n$， $c_1 + c_2 + \ldots + c_m \ge n$

$1 \le n, m \le 500$

(1) 、$n \le 10$

(2) 、$c_1 + c_2 + \ldots + c_m = n$

(3) 、$m = 2$

「CTSC2016」萨菲克斯・阿瑞

求出所有由 m 种字符组成，其中第 i 种字符出现次数不超过 c_i，且长度为 n 的字符串，共有多少种不同的后缀数组。答案对 10^9+7 取模。

$0 \le c_1, c_2, \ldots, c_m \le n$， $c_1+c_2+\ldots+c_m \ge n$

$1 \le n, m \le 500$

(1) 、$n \le 10$

(2) 、$c_1+c_2+\ldots+c_m = n$

(3) 、$m=2$

(4) 、$m=3$

「CTSC2016」萨菲克斯・阿瑞

求出所有由 m 种字符组成，其中第 i 种字符出现次数不超过 c_i，且长度为 n 的字符串，共有多少种不同的后缀数组。答案对 10^9+7 取模。

$0 \le c_1, c_2, \ldots, c_m \le n$， $c_1 + c_2 + \ldots + c_m \ge n$

$1 \le n, m \le 500$

(1) 、$n \le 10$

(2) 、$c_1 + c_2 + \ldots + c_m = n$

(3) 、$m = 2$

(4) 、$m = 3$

(5) 、$1 \le n, m \le 50$

$n \leq 10$

首先考虑一个问题的暴力解法，枚举字符串的后缀数组，判断是否存在一个对应的字符串。

$n \leq 10$

首先考虑一个问题的暴力解法，枚举字符串的后缀数组，判断是否存在一个对应的字符串。

令所枚举的后缀数组为 $p_1, p_2, \ldots, p_n$ ，则显然有

$$s_{p_1} \leq s_{p_2} \leq \cdots \leq s_{p_n}$$

$n \le 10$

首先考虑一个问题的暴力解法，枚举字符串的后缀数组，判断是否存在一个对应的字符串。

令所枚举的后缀数组为 $p_1, p_2, \dots, p_n$ ，则显然有

$$s_{p_1} \le s_{p_2} \le \cdots \le s_{p_n}$$

更进一步地，考虑 $s_{p_i}, s_{p_{i+1}}$ 的大小关系，若 $p_i + 1$ 早于 $p_{i+1} + 1$ 出现，即后缀 $suf[p_i + 1]$ 的字典序要小于后缀 $suf[p_{i+1} + 1]$ ，则 $s_{p_i} \le s_{p_{i+1}}$ ，否则，应当有 $s_{p_i} < s_{p_{i+1}}$ 。

$n \le 10$

首先考虑一个问题的暴力解法，枚举字符串的后缀数组，判断是否存在一个对应的字符串。

令所枚举的后缀数组为 $p_1, p_2, \ldots, p_n$ ，则显然有

$$s_{p_1} \le s_{p_2} \le \cdots \le s_{p_n}$$

更进一步地，考虑 $s_{p_i}, s_{p_{i+1}}$ 的大小关系，若 $p_i + 1$ 早于 $p_{i+1} + 1$ 出现，即后缀 $suf[p_i + 1]$ 的字典序要小于后缀 $suf[p_{i+1} + 1]$ ，则 $s_{p_i} \le s_{p_{i+1}}$ ，否则，应当有 $s_{p_i} < s_{p_{i+1}}$ 。

由此，对于各个字符大小关系的限制应当写作

$$s_{p_1} \oplus s_{p_2} \oplus \cdots \oplus s_{p_n}$$

$n \le 10$

首先考虑一个问题的暴力解法，枚举字符串的后缀数组，判断是否存在一个对应的字符串。

令所枚举的后缀数组为 $p_1, p_2, \dots, p_n$ ，则显然有

$$s_{p_1} \le s_{p_2} \le \cdots \le s_{p_n}$$

更进一步地，考虑 $s_{p_i}, s_{p_{i+1}}$ 的大小关系，若 $p_i + 1$ 早于 $p_{i+1} + 1$ 出现，即后缀 $suf[p_i + 1]$ 的字典序要小于后缀 $suf[p_{i+1} + 1]$ ，则 $s_{p_i} \le s_{p_{i+1}}$ ，否则，应当有 $s_{p_i} < s_{p_{i+1}}$ 。

由此，对于各个字符大小关系的限制应当写作

$$s_{p_1} \oplus s_{p_2} \oplus \cdots \oplus s_{p_n}$$

其中 $\oplus$ 或是 $\le$ ，或是 $<$ 。可以发现，这样的限制是充分且必要的。

$n \leq 10$

那么，判断是否存在一个对应的字符串的过程可以通过贪心来处理。考虑按照上述不等式链上从左向右的顺序填入字符，将最小的字符填入第一个位置，若 s_{p_1} 后的不等号为 $\leq$ ，则重复这个过程，否则，丢弃掉所有相同的字符后，重复这个过程。

$n \le 10$

那么，判断是否存在一个对应的字符串的过程可以通过贪心来处理。考虑按照上述不等式链上从左向右的顺序填入字符，将最小的字符填入第一个位置，若 s_{p_1} 后的不等号为 $\le$ ，则重复这个过程，否则，丢弃掉所有相同的字符后，重复这个过程。

由此，我们得到了问题的一个 $O(n \times n!)$ 的解法。

$c_1 + c_2 + \ldots + c_m = n$

考虑如何对原题计数，首先考虑特殊情况 $\sum c_i = n$ ，此时，答案为

$$\frac{n!}{c_1!c_2!\ldots c_m!}$$

$c_1 + c_2 + \ldots + c_m = n$

考虑如何对原题计数，首先考虑特殊情况 $\sum c_i = n$ ，此时，答案为

$$\frac{n!}{c_1!c_2!\ldots c_m!}$$

这是因为对于任意两个由数量一定的字符组成的字符串，由 1 开头的、由 1, 2 开头的、由 1, 2, 3 开头的……后缀的集合中，一定有一个是对应不同的。因此产生的后缀数组一定是不同的。

$m = 2$

考虑 $m = 2$ 的情况，考虑枚举 $a_1 \le c_1, a_2 \le c_2, a_1 + a_2 = n$，并对 $\frac{n!}{a_1!a_2!}$ 求和。我们发现得到的答案并不正确，这是因为在 $\frac{n!}{a_1!a_2!}$ 中，我们一定会统计到后缀数组 $[n, n-1, \ldots, 2, 1]$ 的存在，从而在重复的求和中，其被计算了多次。

$m = 2$

考虑 $m = 2$ 的情况，考虑枚举 $a_1 \le c_1, a_2 \le c_2, a_1 + a_2 = n$ ，并对 $\frac{n!}{a_1!a_2!}$ 求和。我们发现得到的答案并不正确，这是因为在 $\frac{n!}{a_1!a_2!}$ 中，我们一定会统计到后缀数组 $[n, n-1, \ldots, 2, 1]$ 的存在，从而在重复的求和中，其被计算了多次。

这启示我们在考虑由两种字符组成的字符串的后缀数组时，应当考虑仅计算上述不等式链中，存在小于号的情况。对于不存在小于号的情况，我们可以只使用一种字符构造出来。形式化地来说，我们希望求出 $f(i)$ ，表示有多少可以构造出的后缀数组的不等式链中，存在恰好 $i-1$ 个小于号，也即在字符数量充足的情况下，至少需要使用 i 种不同的字符。求出 $f(i)$ 后，答案显然为

$$\sum f(i)$$

$m = 2, 3$

对于 $m = 2$ 的情况，在枚举 $a_1 \le c_1, a_2 \le c_2, a_1 + a_2 = n$ 后，$f(2)$ 应当等于对 $\frac{n!}{a_1!a_2!} - 1$ 求和的结果。其中减去的 1 就是强制不等式链中不存在小于号，将 a_1, a_2 看做同种字符后，计算得到的后缀数组数 $\frac{n!}{(a_1+a_2)!}$，也即是 $\frac{n!}{n!} = 1$。

$m = 2, 3$

对于 $m = 2$ 的情况，在枚举 $a_1 \le c_1, a_2 \le c_2, a_1 + a_2 = n$ 后，$f(2)$ 应当等于对 $\frac{n!}{a_1!a_2!} - 1$ 求和的结果。其中减去的 1 就是强制不等式链中不存在小于号，将 a_1, a_2 看做同种字符后，计算得到的后缀数组数 $\frac{n!}{(a_1+a_2)!}$ ，也即是 $\frac{n!}{n!} = 1$ 。

由此，我们看到了通过容斥原理解决算重问题的模型。令 $\sum a_i = n$ ，考虑如何计算 $f(m)$ 。则应当是将不等式链的 k 个小于号强制取等，将方案数乘以容斥系数 $(-1)^k$ ，计入答案。

$1 \leq n, m \leq 50$

得到容斥原理计数的思想后，我们便可以着手考虑原问题了。

$1 \leq n, m \leq 50$

得到容斥原理计数的思想后，我们便可以着手考虑原问题了。
考虑枚举 $a_1, a_2, \ldots, a_k$ ($\sum a_i = n$)，则我们需要进行的步骤如下：

$1 \leq n, m \leq 50$

得到容斥原理计数的思想后，我们便可以着手考虑原问题了。
考虑枚举 $a_1, a_2, \ldots, a_k$ $(\sum a_i = n)$ ，则我们需要进行的步骤如下：
(1) 、通过贪心判断不等式链

$$s_{p_1} \leq s_{p_2} \leq \cdots \leq s_{p_{a_1}} < s_{p_{a_1}} \leq s_{p_{a_1+1}} \leq \cdots \leq s_{p_{a_1+a_2}} < \ldots$$

是否可能在字符数限制下得到满足，注意这里具体的后缀数组 p_i 是不重要的，因为对于一个确定的，不存在 $\leq$ ，而仅存在 $<, =$ ，且相邻的 $<$ 之间的元素已经确定的不等式链，一定存在唯一的后缀数组 p_i 与之对应

$1 \le n, m \le 50$

得到容斥原理计数的思想后，我们便可以着手考虑原问题了。
考虑枚举 $a_1, a_2, \dots, a_k$ ($\sum a_i = n$)，则我们需要进行的步骤如下：
(1)、通过贪心判断不等式链

$$s_{p_1} \le s_{p_2} \le \dots \le s_{p_{a_1}} < s_{p_{a_1}} \le s_{p_{a_1+1}} \le \dots \le s_{p_{a_1+a_2}} < \dots$$

是否可能在字符数限制下得到满足，注意这里具体的后缀数组 p_i 是不重要的，因为对于一个确定的，不存在 $\le$，而仅存在 $<, =$，且相邻的 $<$ 之间的元素已经确定的不等式链，一定存在唯一的后缀数组 p_i 与之对应
(2)、对于可能得到满足的 $\{a_i\}$，计算 $f(k)$，统计入答案

$1 \leq n, m \leq 50$

得到容斥原理计数的思想后，我们便可以着手考虑原问题了。
考虑枚举 $a_1, a_2, \ldots, a_k$ ($\sum a_i = n$)，则我们需要进行的步骤如下：
(1)、通过贪心判断不等式链

$$s_{p_1} \leq s_{p_2} \leq \cdots \leq s_{p_{a_1}} < s_{p_{a_1}} \leq s_{p_{a_1+1}} \leq \cdots \leq s_{p_{a_1+a_2}} < \ldots$$

是否可能在字符数限制下得到满足，注意这里具体的后缀数组 p_i 是不重要的，因为对于一个确定的，不存在 $\leq$，而仅存在 $<, =$，且相邻的 $<$ 之间的元素已经确定的不等式链，一定存在唯一的后缀数组 p_i 与之对应

(2)、对于可能得到满足的 $\{a_i\}$，计算 $f(k)$，统计入答案

考虑设计动态规划来加速上述过程，令 $dp_{i,j,k}$ 表示当前考虑了 i 种字符，对于已经确定的 $\{a_i\}$，有 $\sum a_i = j$，且 $\{a_i\}$ 的最后一项为 k 的情况下，乘上容斥系数的方案数和。

$1 \leq n, m \leq 50$

转移分为如下三类：

$1 \le n, m \le 50$

转移分为如下三类：

(1) 、使用若干个第 i 种字符后，不等式链上出现一个小于号，不强制取等。枚举第 i 种字符使用的个数 $x\,(1 \le x \le c_i)$ ，则有

$$dp_{i+1,j+x,0} \Leftarrow \frac{1}{(k+x)!} \times dp_{i,j,k}$$

$1 \leq n, m \leq 50$

转移分为如下三类：

(1)、使用若干个第 i 种字符后，不等式链上出现一个小于号，不强制取等。枚举第 i 种字符使用的个数 $x\,(1 \leq x \leq c_i)$，则有

$$dp_{i+1,j+x,0} \Leftarrow \frac{1}{(k+x)!} \times dp_{i,j,k}$$

(2)、使用若干个第 i 种字符后，不等式链上出现一个小于号，强制取等。枚举第 i 种字符使用的个数 $x\,(1 \leq x \leq c_i)$，则有

$$dp_{i+1,j+x,k+x} \Leftarrow (-1) \times dp_{i,j,k}$$

$1 \le n, m \le 50$

转移分为如下三类：

(1)、使用若干个第 i 种字符后，不等式链上出现一个小于号，不强制取等。枚举第 i 种字符使用的个数 $x\,(1 \le x \le c_i)$，则有

$$dp_{i+1,j+x,0} \Leftarrow \frac{1}{(k+x)!} \times dp_{i,j,k}$$

(2)、使用若干个第 i 种字符后，不等式链上出现一个小于号，强制取等。枚举第 i 种字符使用的个数 $x\,(1 \le x \le c_i)$，则有

$$dp_{i+1,j+x,k+x} \Leftarrow (-1) \times dp_{i,j,k}$$

(3)、用尽全部的第 i 种字符，不等式链上仍未出现小于号，则有

$$dp_{i+1,j+c_i,k+c_i} \Leftarrow dp_{i,j,k}$$

$1 \leq n, m \leq 50$

由此，我们得到了一个利用动态规划计数的解法。

$1 \le n, m \le 50$

由此，我们得到了一个利用动态规划计数的解法。
时间复杂度 $O(n^3 \times m)$ 。

$1 \le n, m \le 500$

观察上述动态规划的复杂度瓶颈 (1), (2) ，不难发现其转移可以通过斜向前缀和进行优化。

$1 \le n, m \le 500$

观察上述动态规划的复杂度瓶颈 $(1), (2)$ ，不难发现其转移可以通过斜向前缀和进行优化。

优化后的时间复杂度为 $O(n^2 \times m)$ 。

Thanks

感谢大家的聆听。