试题精讲及拓展

Arraiter

2023 年 11 月 9 日

吐槽

本场难度基本上对标 NOIP2021。
原因是在参与 NOIP2021 中深刻感受到了在思维题与大数据结构题之间取舍的痛苦。
T1 很简单，有一点易错点。
T2 很一眼，有一点细节。
T3 是大数据结构，但是容易想到，对标 NOIP2021 T4，但应该简单很多。
T4 比较思维，对标 NOIP2021 T3，但应该困难一点。

题目大意

在给定集合中找到和相等的两个不交子集。
集合元素形如 2^{x_i}。
$n, x_i \leq 1000$

得分情况与吐槽

题目分析

如果没有元素形如 2^{x_i} 的保证，这是经典的不可做问题。

题目分析

如果没有元素形如 2^{x_i} 的保证，这是经典的不可做问题。
通过二进制的性质解决问题。

题目分析：子任务 4

注意到 n 的范围远大于 x_i 的范围。
若 $n > 33$ 由抽屉原理，必然存在 $i \neq j, x_i = x_j$。令 $p = \{i\}, q = \{j\}$ 即可得到一组解。

题目分析：子任务 4

注意到 n 的范围远大于 x_i 的范围。
若 $n > 33$ 由抽屉原理，必然存在 $i \neq j, x_i = x_j$。令 $p = \{i\}, q = \{j\}$ 即可得到一组解。
只需解决 $n \leq 33, x_i \leq 32$ 的子问题。

题目分析

结论：若不存在相等元素，原问题无解。

题目分析

结论：若不存在相等元素，原问题无解。
证明：每个数存在唯一的二进制表示。

题目大意

记 $f(S)$ 为括号序列 S 中合法子序列的个数。
求

$$(\sum_{l=1}^{n}\sum_{r=l}^{n}(l \oplus r \oplus f(S[l,r]))) \bmod 998244353$$

$n \leq 500$

得分情况与吐槽

题目分析

动态规划

不妨将 $f(S[l,r])$ 记作 $f(l,r)$

题目分析

动态规划

不妨将 $f(S[l, r])$ **记作** $f(l, r)$

若 $S_l =$ ）， **有** $f(l, r) = f(l+1, r)$

题目分析

动态规划

不妨将 $f(S[l,r])$ **记作** $f(l,r)$

若 $S_l=$ ），**有** $f(l,r)=f(l+1,r)$

若 $S_l=$（，**枚举与** S_l **配对的右括号** $S_k=$ ），**有**

$$f(l,r)=f(l+1,r)+\sum_{S_k=)} f(l+1,k-1)f(k+1,r)$$

题目分析：异或

$$(\sum_{l=1}^{n}\sum_{r=l}^{n}(l \oplus r \oplus f(S[l,r]))) \bmod 998244353$$

题目分析：异或

$$\left(\sum_{l=1}^{n}\sum_{r=l}^{n}(l \oplus r \oplus f(S[l,r]))\right) \bmod 998244353$$

异或的性质：

a=011101

b=11010010101100...

c=10100110101100...

题目分析：异或

$$(\sum_{l=1}^{n}\sum_{r=l}^{n}(l \oplus r \oplus f(S[l,r]))) \bmod 998244353$$

异或的性质：

a=011101

b=11010010101100...

c=10100110101100...

$a \oplus b$ 仅与 a, b 的低位有关

题目分析

根据转移公式计算 $f(S[l,r]) \bmod 512$ 和 $f(S[l,r]) \bmod 998244353$
通过比较 $l, r, f(S[l,r]) \bmod 512$ 的低位，计算 $(l \oplus r \oplus f(S[l,r])) - f(S[l,r])$
时间复杂度 $O(n^3)$

题目大意

给定有向图，每次操作可以合并两个点，求最少操作多少次使得图变为一张可以有重边的有向环。
$n \leq 10^5$，$m \leq 2 \times 10^5$

得分情况与吐槽

题目分析：子任务 2

问题看作最大化新图的点数 n'。下文称答案为 n' 的最大值。

题目分析：子任务 2

问题看作最大化新图的点数 n'。下文称答案为 n' 的最大值。
记点 u 在新图中的标号为 a_u，原问题中的缩点等价于给出新的标号。

题目分析：子任务 2

问题看作最大化新图的点数 n'。下文称答案为 n' 的最大值。
记点 u 在新图中的标号为 a_u，原问题中的缩点等价于给出新的标号。
则：

- $(u, v) \in G \implies (a_u, a_v) \in G'$
- $(u, v), (v, w) \in G \implies (a_u, a_v) \in G', (a_v, a_w) \in G'$
- $\cdots$
- 任意原图 G 中的路径 $p_1, p_2, \cdots, p_k$，在新图中也构成一条路径 $a_{p_1}, a_{p_2}, \cdots, a_{p_k}$

题目分析：子任务 2

问题看作最大化新图的点数 n'。下文称答案为 n' 的最大值。
记点 u 在新图中的标号为 a_u，原问题中的缩点等价于给出新的标号。
则：

- $(u, v) \in G \implies (a_u, a_v) \in G'$
- $(u, v), (v, w) \in G \implies (a_u, a_v) \in G', (a_v, a_w) \in G'$
- $\cdots$
- 任意原图 G 中的路径 $p_1, p_2, \cdots, p_k$，在新图中也构成一条路径 $a_{p_1}, a_{p_2}, \cdots, a_{p_k}$

任意原图中的环 $p_1, p_2, \cdots, p_k, p_1 = p_k$，在新图中也构成一个环（不一定是简单环）$a_{p_1}, a_{p_2}, \cdots, a_{p_k}, a_{p_1} = a_{p_k}$

题目分析：子任务 2

问题看作最大化新图的点数 n'。下文称答案为 n' 的最大值。
记点 u 在新图中的标号为 a_u，原问题中的缩点等价于给出新的标号。
则：

- $(u, v) \in G \implies (a_u, a_v) \in G'$
- $(u, v), (v, w) \in G \implies (a_u, a_v) \in G', (a_v, a_w) \in G'$
- $\cdots$
- 任意原图 G 中的路径 $p_1, p_2, \cdots, p_k$，在新图中也构成一条路径 $a_{p_1}, a_{p_2}, \cdots, a_{p_k}$

任意原图中的环 $p_1, p_2, \cdots, p_k, p_1 = p_k$，在新图中也构成一个环（不一定是简单环）
$a_{p_1}, a_{p_2}, \cdots, a_{p_k}, a_{p_1} = a_{p_k}$
故 n' 是 k 的约数。

题目分析：子任务 2

问题看作最大化新图的点数 n'。下文称答案为 n' 的最大值。
记点 u 在新图中的标号为 a_u，原问题中的缩点等价于给出新的标号。
则：

- $(u, v) \in G \implies (a_u, a_v) \in G'$
- $(u, v), (v, w) \in G \implies (a_u, a_v) \in G', (a_v, a_w) \in G'$
- $\cdots$
- 任意原图 G 中的路径 $p_1, p_2, \cdots, p_k$，在新图中也构成一条路径 $a_{p_1}, a_{p_2}, \cdots, a_{p_k}$

任意原图中的环 $p_1, p_2, \cdots, p_k, p_1 = p_k$，在新图中也构成一个环（不一定是简单环）$a_{p_1}, a_{p_2}, \cdots, a_{p_k}, a_{p_1} = a_{p_k}$
故 n' 是 k 的约数。
令 $a_u = rk(u) \bmod \gcd\{k\}$，其中 $rk(u)$ 是 u 在环上的编号 $(1, 2, \cdots, k)$。则 a_u 是一组合法的标号。

题目分析：子任务 2

问题看作最大化新图的点数 n'。下文称答案为 n' 的最大值。
记点 u 在新图中的标号为 a_u，原问题中的缩点等价于给出新的标号。
则：

- $(u, v) \in G \implies (a_u, a_v) \in G'$
- $(u, v), (v, w) \in G \implies (a_u, a_v) \in G', (a_v, a_w) \in G'$
- $\cdots$
- 任意原图 G 中的路径 $p_1, p_2, \cdots, p_k$，在新图中也构成一条路径 $a_{p_1}, a_{p_2}, \cdots, a_{p_k}$

任意原图中的环 $p_1, p_2, \cdots, p_k, p_1 = p_k$，在新图中也构成一个环（不一定是简单环）
$a_{p_1}, a_{p_2}, \cdots, a_{p_k}, a_{p_1} = a_{p_k}$
故 n' 是 k 的约数。
令 $a_u = rk(u) \bmod \gcd\{k\}$，其中 $rk(u)$ 是 u 在环上的编号 $(1, 2, \cdots, k)$。则 a_u 是一组合法的标号。
故答案为所有环长的 $\gcd$。

题目分析

- $(u,v),(u,w)\in G \implies (a_u,a_v),(a_u,a_w)\in G \implies a_v=a_w$

题目分析

- $(u, v), (u, w) \in G \implies (a_u, a_v), (a_u, a_w) \in G \implies a_v = a_w$

若 u 存在两条出边，指向不同的点 v, w，可以用一次操作合并 v, w。

题目分析

- $(u, v), (u, w) \in G \implies (a_u, a_v), (a_u, a_w) \in G \implies a_v = a_w$

若 u 存在两条出边，指向不同的点 v, w，可以用一次操作合并 v, w。

- $(v, u), (w, u) \in G \implies (a_v, a_u), (a_w, a_u) \in G \implies a_v = a_w$

若 u 存在两条入边，由不同的点 v, w 出发，可以用一次操作合并 v, w。

题目分析

- $(u, v), (u, w) \in G \implies (a_u, a_v), (a_u, a_w) \in G \implies a_v = a_w$

若 u 存在两条出边，指向不同的点 v, w，可以用一次操作合并 v, w。

- $(v, u), (w, u) \in G \implies (a_v, a_u), (a_w, a_u) \in G \implies a_v = a_w$

若 u 存在两条入边，由不同的点 v, w 出发，可以用一次操作合并 v, w。
原图可以被化简使得：每个点只有至多一条入边和出边。

题目分析

- $(u, v), (u, w) \in G \implies (a_u, a_v), (a_u, a_w) \in G \implies a_v = a_w$

若 u 存在两条出边，指向不同的点 v, w，可以用一次操作合并 v, w。

- $(v, u), (w, u) \in G \implies (a_v, a_u), (a_w, a_u) \in G \implies a_v = a_w$

若 u 存在两条入边，由不同的点 v, w 出发，可以用一次操作合并 v, w。
原图可以被化简使得：每个点只有至多一条入边和出边。
即简化后的图由有向环和有向链构成。

题目分析

若简化后的图存在环，答案有上界 $\gcd\{k\}$，其中 $\{k\}$ 是环长的集合。同时该上界可以取到。

题目分析

若简化后的图存在环，答案有上界 $\gcd\{k\}$，其中 $\{k\}$ 是环长的集合。同时该上界可以取到。否则，简化后的图只有有向链，将这些链首尾相连即可，答案为 $n -$ 链长之和。

优化合并点的过程即可通过此题。

优化合并点的过程即可通过此题。
但存在更好实现的做法。

优化合并点的过程即可通过此题。
但存在更好实现的做法。
若简化后的图存在环，新建一张带权的有向图 H，对于 $(u,v) \in G$，新建 $(u,v,1) \in H$ 和 $(u,v,-1) \in H$，答案为这张图的所有环长的 gcd。

优化合并点的过程即可通过此题。
但存在更好实现的做法。
若简化后的图存在环，新建一张带权的有向图 H，对于 $(u,v) \in G$，新建 $(u,v,1) \in H$ 和 $(u,v,-1) \in H$，答案为这张图的所有环长的 gcd。
否则，原图是弱连通森林。可以发现，经过上述化简过程，一个弱连通树等价于这棵树内最长的有向路径。

优化合并点的过程即可通过此题。
但存在更好实现的做法。
若简化后的图存在环，新建一张带权的有向图 H，对于 $(u,v)\in G$，新建 $(u,v,1)\in H$ 和 $(u,v,-1)\in H$，答案为这张图的所有环长的 gcd。
否则，原图是弱连通森林。可以发现，经过上述化简过程，一个弱连通树等价于这棵树内最长的有向路径。
对每个连通块求内部最长路径即可。

题目大意

将数码字符串 a 划分为若干非空子串，然后将每个非空子串视为一个数（可能有前导零）按照在 a 中的顺序写下来，记得到的序列为 $\{b_1, \cdots, b_k\}$，求出满足以下条件的序列 b:

- 对于所有 $1 \le i < k$，有 $b_i < b_{i+1}$；
- b_k 尽可能小；
- 在满足上一条限制的前提下，序列 b 的字典序尽可能大。

$n \le 10^5$

得分情况

题目分析：子任务 2

记 $num(l,r)$ 为子串 $S[l,r]$ 对应的数。记 $len(x)$ 为数 x 长度，特别的 $len(num(l,r))$ 是 $r-l+1-$ (num(l,r) 的前导零个数)。

考虑动态规划

题目分析：子任务 2

记 $num(l,r)$ **为子串** $S[l,r]$ **对应的数。记** $len(x)$ **为数** x **长度，特别的** $len(num(l,r))$ **是**
$r-l+1-$ (`num(l,r)` 的前导零个数)。
考虑动态规划
记 $f_{i,j}$ **为，仅考虑** a **的前** i **位，将区间** $[j,i]$ **划分为一个子串，序列** b **的字典序最大是多少。**

题目分析：子任务 2

记 $num(l,r)$ **为子串** $S[l,r]$ **对应的数。记** $len(x)$ **为数** x **长度，特别的** $len(num(l,r))$ **是** $r-l+1-$ (`num(l,r)` 的前导零个数)。

考虑动态规划

记 $f_{i,j}$ **为，仅考虑** a **的前** i **位，将区间** $[j,i]$ **划分为一个子串，序列** b **的字典序最大是多少。**

若区间 $[j,i]$ **对应的数比区间** $[i+1,k]$ **的数小，则可以从** $f_{i,j}$ **转移到** $f_{k,i+1}$。

题目分析：子任务 3

我们的按位比较中，仅有第二条限制会尽量选更小的 b_i，其余都会选较大的。

题目分析：子任务 3

我们的按位比较中，仅有第二条限制会尽量选更小的 b_i，其余都会选较大的。
注意到 b_k 尽可能小，等价于划分出来的最后一段的长度最小。

题目分析：子任务 3

我们的按位比较中，仅有第二条限制会尽量选更小的 b_i，其余都会选较大的。
注意到 b_k 尽可能小，等价于划分出来的最后一段的长度最小。
考虑先求出 b_k 由哪一段划分构成。

题目分析：子任务 3

我们的按位比较中，仅有第二条限制会尽量选更小的 b_i，其余都会选较大的。
注意到 b_k 尽可能小，等价于划分出来的最后一段的长度最小。
考虑先求出 b_k 由哪一段划分构成。
沿用上述状态 $f_{i,j}$。

题目分析：子任务 3

我们的按位比较中，仅有第二条限制会尽量选更小的 b_i，其余都会选较大的。
注意到 b_k 尽可能小，等价于划分出来的最后一段的长度最小。
考虑先求出 b_k 由哪一段划分构成。
沿用上述状态 $f_{i,j}$。
对于 $k < j$，若 $f_{i,j}$ 和 $f_{i,k}$ 同时存在，则从 $f_{i,j}$ 转移过来一定不劣。

题目分析：子任务 3

我们的按位比较中，仅有第二条限制会尽量选更小的 b_i，其余都会选较大的。
注意到 b_k 尽可能小，等价于划分出来的最后一段的长度最小。
考虑先求出 b_k 由哪一段划分构成。
沿用上述状态 $f_{i,j}$。
对于 $k < j$，若 $f_{i,j}$ 和 $f_{i,k}$ 同时存在，则从 $f_{i,j}$ 转移过来一定不劣。
将第二维省略，记 f_i 为能够使 $f_{i,j}$ 存在的最大的 j。
有转移

$$f_i = \max\{j + 1 | j < i, num(f_j, j) < num(j + 1, i)\}$$

题目分析：子任务 3

我们的按位比较中，仅有第二条限制会尽量选更小的 b_i，其余都会选较大的。
注意到 b_k 尽可能小，等价于划分出来的最后一段的长度最小。
考虑先求出 b_k 由哪一段划分构成。
沿用上述状态 $f_{i,j}$。
对于 $k < j$，若 $f_{i,j}$ 和 $f_{i,k}$ 同时存在，则从 $f_{i,j}$ 转移过来一定不劣。
将第二维省略，记 f_i 为能够使 $f_{i,j}$ 存在的最大的 j。
有转移

$$f_i = \max\{j+1 | j < i, num(f_j, j) < num(j+1, i)\}$$

计算完 f_n 后，我们有 $b_k = num(f_n, n)$。

题目分析：子任务 3

类似的，记 $g_{i,j}$ 为，仅考虑 $a[i,n]$，将 $a[i,j]$ 划分为一段，得到的最小的 b。

题目分析：子任务 3

类似的，记 $g_{i,j}$ 为，仅考虑 $a[i,n]$，将 $a[i,j]$ 划分为一段，得到的最小的 b。
沿用上述想法，令 g_i 为，能够使得 $g_{i,j}$ 存在的最小的 j，有转移

$$g_i = \min\{j-1 | i < j, num(i,j-1) < num(j,g_j)\}$$

题目分析：子任务 3

类似的，记 $g_{i,j}$ 为，仅考虑 $a[i,n]$，将 $a[i,j]$ 划分为一段，得到的最小的 b。
沿用上述想法，令 g_i 为，能够使得 $g_{i,j}$ 存在的最小的 j，有转移

$$g_i = \min\{j-1 | i < j, num(i,j-1) < num(j,g_j)\}$$

时间复杂度 $O(n^2)$

题目分析

优化转移:

- $f_i = \max\{j+1 | j < i, num(f_j, j) < num(j+1, i)\}$
- $g_i = \min\{j-1 | i < j, num(i, j-1) < num(j, g_j)\}$

题目分析

优化转移：

- $f_i = \max\{j+1 | j < i, num(f_j, j) < num(j+1, i)\}$
- $g_i = \min\{j-1 | i < j, num(i, j-1) < num(j, g_j)\}$

对于 f 的转移：枚举 j，确定 f_j 后，$num(f_j, j) < num(j+1, i)$ 对一段后缀的 i 成立。若 $len(num(f_j, j)) < len(num(j+1, i))$，必然可以转移；当 $len(num(f_j, j)) = len(num(j+1, i))$ 时，问题等价于比较两个长度相等的子串的字典序，可以通过二分 + 哈希或者后缀数组实现。

题目分析

优化转移：

- $f_i = \max\{j+1 | j < i, num(f_j, j) < num(j+1, i)\}$
- $g_i = \min\{j-1 | i < j, num(i, j-1) < num(j, g_j)\}$

对于 f 的转移：枚举 j，确定 f_j 后，$num(f_j, j) < num(j+1, i)$ 对一段后缀的 i 成立。若 $len(num(f_j, j)) < len(num(j+1, i))$，必然可以转移；当 $len(num(f_j, j)) = len(num(j+1, i))$ 时，问题等价于比较两个长度相等的子串的字典序，可以通过二分 + 哈希或者后缀数组实现。

对于 g 的转移：枚举 j，确定 f_j 后，满足 $num(i, j-1) < num(j, g_j)$ 的 i 构成一段以 $j-1$ 为右端点的区间，可以类似求出，并用单调队列辅助转移。

题目分析

优化转移：

- $f_i = \max\{j+1 | j < i, num(f_j, j) < num(j+1, i)\}$
- $g_i = \min\{j-1 | i < j, num(i, j-1) < num(j, g_j)\}$

对于 f 的转移：枚举 j，确定 f_j 后，$num(f_j, j) < num(j+1, i)$ 对一段后缀的 i 成立。若 $len(num(f_j, j)) < len(num(j+1, i))$，必然可以转移；当 $len(num(f_j, j)) = len(num(j+1, i))$ 时，问题等价于比较两个长度相等的子串的字典序，可以通过二分 + 哈希或者后缀数组实现。

对于 g 的转移：枚举 j，确定 f_j 后，满足 $num(i, j-1) < num(j, g_j)$ 的 i 构成一段以 $j-1$ 为右端点的区间，可以类似求出，并用单调队列辅助转移。

时间复杂度 $O(n \log n)$，瓶颈在于 $O(n)$ 次比较最长公共前缀。