解题报告 by 雅礼中学 邓宇帆

QOJ850 Edit Distance Yet Again

题目大意

给定两个字符串 S 和 T 以及参数 k, 求它们之间的编辑距离与 k 的较小值. 如果编辑距离不超过 k, 则输出任意一种方案. 编辑距离定义为通过对其中一个字符串进行最少的 插入一个字符, 删除一个字符, 修改一个字符 操作, 使得其变为另一个字符串的操作次数.

数据范围

令 t 为数据组数, n 为 S 的长度, m 为 T 的长度, N 为每组数据中 S 的长度之和, M 为每组数据中 T 的长度之和.

$1 \leq t \leq 100, 1 \leq n, m \leq 10^6, 0 \leq k \leq 1000$.

$1 \leq N, M \leq 10^7$.

解题过程

先不考虑构造操作方案.

显然有一个 $O(nm)$ 的 Dp, 即设 $f_{i,j}$ 表示使 S 中长度为 i 的前缀与 T 中长度为 j 的前缀变得相同的最少操作次数.

转移考虑三种情况, 也即如果接下来进行一次插入操作, 则有 $f_{i,j+1} \leftarrow f_{i,j} + 1$; 如果接下来进行一次删除操作, 则有 $f_{i+1,j} \leftarrow f_{i,j} + 1$; 如果接下来进行一次修改操作 (前提是需要修改), 则有 $f_{i+1,j+1} \leftarrow f_{i,j} + [s_{i+1} \neq t_{j+1}]$. 观察转移, 注意到如果 $i - j$ 发生变化, 一定会伴随着 Dp 值增加 1.

进一步, 有一个显然的优化, 也即 i 和 j 的差距如果大于 k, 则 Dp 值也必然大于 k, 可以将这些状态删去, 则可设 $g_{i,d}$ 表示使 S 中长度为 i 的前缀与 T 中长度为 $(i + d)$ 的前缀变得相同的最少操作次数, 转移是类似的. 注意到 $|d| \leq k$, 故状态数变为 $O(nk)$.

这启发我们继续将 i 一维变为 $O(k)$ 量级. 注意到 Dp 值不会大于 k, 不难想到进行 Dp of Dp. 但我们需要找到一个 Dp 值关于 i 的单调性.

考虑一对 i_1 和 i_2, 有 $i_1 < i_2$. 我们声称若 $g_{i_1,d} \geq g_{i_2,d}$, 则对于全局的 S 和 T 的最优策略, 如果没有经过状态 (i_2, d), 就一定不会经过状态 (i_1, d). 换言之, 不需要对 $g_{i_1,d}$ 进行转移.

考虑如果最优策略经过状态 (i_1, d), 在此之后进行的操作. 记录 t 为状态 (i_1, d) 变为 (i_2, d') $(d' \geq d)$ 或者 $(i_1', i_2 + d - i_1')$ $(i_1' \geq i_2)$ 的时刻. 形象地描述, 即为第一次严格超过 (i_2, d) 的时刻.

假设此时的状态为 (i_2, d') $(d' \geq d)$, 操作次数为 w, 另一种情况是对称的. 则显然有 $w \geq g_{i_1,d} + (d' - d) \geq g_{i_2,d} + (d' - d)$. 那么考虑从 (i_2, d) 开始操作, 进行 $(d' - d)$ 次插入操作, 则状态同样变为 (i_2, d'), 且有 $g_{i_2,d} + (d' - d) \leq w$. 故经过状态 (i_2, d) 不劣于经过状态 (i_1, d).

于是我们可以愉快地进行 Dp of Dp. 设 $h_{w,d}$ 表示 $g_{i,d} \leq w$ 情况下 i 的最大值, 则状态数变为 $O(k^2)$. 转移同样不难, 唯一需要注意的是在进行了 插入/删除/修改 操作之后, 还要尽量的往后匹配一段相同的子串. 这部分可以使用二分+哈希做到 $O(\log n)$, 当然也可以用 SA 做到 $O(1)$.

构造方案只需要在得到所有的 Dp 值后从终止状态逆推回初始状态即可.

如果使用二分+哈希, 单组数据的时间复杂度为 $O(n + k^2 \log n)$, 空间复杂度为 $O(n + k^2)$. 如果使用 SA, 单组数据的时空复杂度均为 $O(n \log n + k^2)$.

参考资料

Petrozavodsk Camp, Day 1: Solutions

QOJ856 Cactus

题目大意

给定一棵仙人掌, 你需要用 k 种颜色给每个结点染色, 且保证有边相连的结点的颜色不相同, 求染色的方案数对 $(10^9 + 7)$ 取模的结果. 仙人掌定义为一张特殊的无向图, 其中每条边至多在一个简单环上.

数据范围

令 t 为数据组数, n 为仙人掌的点数, m 为仙人掌的边数, N 为每组数据中仙人掌的点数之和, M 为每组数据中仙人掌的边数之和.

$1 \leq t \leq 5 \times 10^4$, $1 \leq n \leq 3 \times 10^5$, $0 \leq m \leq 4 \times 10^5$, $2 \leq k \leq 10^9$.

$1 \leq N \leq 3 \times 10^6$, $1 \leq M \leq 4 \times 10^6$.

解题过程

考虑一种特殊情况, 即仙人掌是树时的做法. 显然可以随意确定一个根, 那么除了根以外的结点, 其的颜色不能与父亲相同, 故能取到的颜色数都为 $(k-1)$, 答案为 $k(k-1)^{n-1}$. 这启发我们尝试把仙人掌转化为树的情况.

不难想到, 如果把一个简单环看作一个结点, 那么仙人掌就几乎成为了一颗树. 但由于两个简单环可能会有一个公共结点, 这个转化还显得没有良好定义. 于是我们可以把公共结点拆成若干份, 使得每个简单环都包含其中一份, 并在份与份之间连接特殊边, 使得这些特殊边形成树的结构, 并规定特殊边两侧的结点颜色需要相同. 这样, 我们有一个结点至多在一个简单环内.

这样, 我们就可以将每个简单环看成一个结点, 仙人掌就变成了一棵树. 依旧随意定一个根, 那么除了根以外的结点, 其所代表的结点或简单环内有恰好一个结点要与父亲边另一侧的结点的颜色相同或不同. 我们将一个简单环看作一个黑盒, 由于不同颜色之间的地位相同, 上述限制对简单环内部染色的方案数的影响为将方案数乘以 $\frac{1}{k}$ 或 $\frac{k-1}{k}$.

如果不考虑一个简单环内部的方案数, 此时的答案是容易计算的, 只需统计结点数量, 普通边数量和特殊边数量即可.

故现在我们需要计算一个 s 个结点的简单环的染色方案数. 一个朴素的想法是断环成链. 设 s 个结点的简单环的染色方案数为 f_s, s 个结点的链的染色方案数为 $g_s = k(k-1)^{s-1}$.

我们在环上任取一条边将其断开, 则简单环变为一条长度为 s 的链. 但在此之后需要减去这条边两侧的结点颜色相同的方案数, 也即 f_{s-1}. 故有递推式 $f_s = g_s - f_{s-1}$.

最后, 我们事实上不需要显式地将公共结点拆分, 而只需要统计圆方树上圆点的度数即可. 类似地, 统计环上的结点数也只需考虑方点的度数即可.

时空复杂度均为线性.

参考资料

Petrozavodsk Camp, Day 1: Solutions

QOJ862 Social Justice

题目大意

给定一个长度为 n 的序列 $\{a_i\}$ 和有理数参数 k, 定义一个长度为 m 的子序列 $\{b_i\}$ 是好的当且仅当

$$\frac{\max_{i=1}^{m} b_i}{k} \leq \frac{\sum_{i=1}^{m} b_i}{m}.$$

也即最大值不超过平均值的 k 倍. 求所有长度最大的好的子序列的并集.

数据范围

令 t 为数据组数, k 能被表示为分数 $\frac{p}{q}$, N 为所有数据中 n 的和.

$1 \leq t \leq 1000, 1 \leq n \leq 2 \times 10^5, 0 \leq a_i \leq 10^9, 1 \leq q < p \leq 1000.$

$1 \leq N \leq 10^6.$

解题过程

首先注意到, 题目与 a_i 的顺序无关, 于是自然将 $\{a_i\}$ 升序排序. 为了方便, 我们假定 a_i 两两不同, 这可以通过将 a_i 看作二元组 (a_i, i) 实现, 从而有 $i < j \leftrightarrow a_i < a_j$.

由于需要求所有长度最大的好的子序列的并集, 因此我们需要先求出最大的好的子序列长度.

观察 1. 如果存在一个最大元素为 a_i 的长度为 m 的好的子序列 $(m > 1)$, 则一定存在一个最大元素为 a_i 的长度为 $m-1$ 的好的子序列.

Proof. 将原先的子序列的最小值删去, 显然有最大值不变, 而平均值不减, 故依旧合法.

观察 2. 如果存在一个最大元素为 a_i 的长度为 m 的好的子序列, 则子序列 $\{ a_{i-m+1}, a_{i-m+2}, \cdots, a_i \}$ 是好的. 换言之, 以 a_i 结尾的长度为 m 的区间是好的.

Proof. 考虑原先的子序列, 反复地取出最小值, 将其改为不超过 a_i 的最大的不在当前子序列中的元素, 显然改完之后依旧合法, 且有限次操作后即可变为一个区间.

因此, 我们枚举每个 i 作为右端点, 对左端点进行二分, 即可找到每个右端点最长的区间. 取最长的区间的长度即可.

现在我们需要求长度最大的好的子序列的并集. 我们尝试对每个元素考虑, 能否有一个好的子序列包含了这个元素.

观察 3. 如果存在一个最大元素为 a_i 的长度为 m 的好的子序列包含了元素 a_j, 则若 $j > i - m$, 有 a_j 在以 a_i 结尾的长度为 m 的区间中; 否则 $j \le i - m$, 有 $\{ a_j \} \cup \{ a_{i-m+2}, a_{i-m+3}, \cdots, a_i \}$ 是好的.

Proof. 前一种情况已证明, 考虑后一种情况. 事实上也是类似于观察 2 的证明, 只是选取最小值时, 若最小值是 a_j 则取次小值即可.

因此, 我们依旧枚举每个 i 作为右端点. 设 m 为之前求出的最大的好的子序列长度. 若以 a_i 结尾的长度为 m 的区间不是好的, 则跳过这个 a_i. 否则对于观察 3 的第一种情况, 标记区间内的元素在并集内; 对于第二种情况, 二分找到最小的 a_j 使得上文描述的子序列合法, 标记从 a_j 到 a_{i-m} 的元素在并集内.

对于单组数据, 时间复杂度为 $O(n \log n)$, 空间复杂度为 $O(n)$.

参考资料

Petrozavodsk Camp, Day 1: Solutions

QOJ8190 Jaw-Dropping Set

题目大意

给定正整数 n, 定义集合 $\{ 1, 2, \cdots, n \}$ 的子集 A 是 **好** 的, 当且仅当 $x \in A \land y \in A \land x \neq y \to x \nmid y$, 即元素两两之间没有倍数关系.

定义集合 B 是 **很好** 的, 当且仅当 B 是所有 **好** 的集合中元素数量最多的集合.

定义集合 C 是 **超级好** 的, 当且仅当 C 是所有 **很好** 的集合中元素总和最小的集合.

求任意一个超级好的集合的元素总和.

数据范围

令 t 为数据组数.

$1 \leq t \leq 10^5, 1 \leq n \leq 10^9$.

解题过程

先考虑求得 **很好** 的集合的元素数量, 令这个量为 k. 设 $m = \lfloor \frac{n}{2} \rfloor$, 注意到, 我们可以取 $S = \{ m+1, m+2, \cdots, n \}$, 有 S 是 **很好** 的集合, 且 $|S| = n - m = \lceil \frac{n}{2} \rceil$. 故 $k \geq \lceil \frac{n}{2} \rceil$.

再考虑将 n 以内的所有正整数写成 $(a \times 2^b)$ 的形式, 其中 a 是奇数. 定义若两个数的 a 相同则在同一个等价类内. 不难发现一个 **好** 的集合内的元素一定两两不在同一个等价类内. 又有等价类数量为 n 以内的奇数个数, 即为 $\lceil \frac{n}{2} \rceil$, 故 $k \leq \lceil \frac{n}{2} \rceil$. 综上所述, $k = \lceil \frac{n}{2} \rceil$.

我们沿用上述的等价类的解题工具. 考虑如果有两个数 $d_1 = (a_1 \times 2^{b_1})$ 和 $d_2 = (a_2 \times 2^{b_2})$, 且 $a_1 \mid a_2$, 则 d_1 和 d_2 要同时在一个 **好** 的集合中, 当且仅当 $b_1 > b_2$.

这启发我们考虑不同等价类之间 b 的关系. 定义 b_i 表示 $a = i$ 的等价类中, 保证每个等价类都有一个元素在集合内的情况下, b 的最小值. 显然奇数之间最小的倍数关系是 3 倍, 我们有 $b_i \geq b_{3i} + 1$. 容易看出 $b_i \geq \lfloor \log_3 \frac{n}{i} \rfloor$, 且容易证明这是一个能够取到的下界.

故我们要计算 $\sum_{i=1}^{n} [2 \nmid i] i \times 2^{\lfloor \log_3 \frac{n}{i} \rfloor}$. 我们有

$$\begin{aligned}\sum_{i=1}^{n}[2\nmid i]i\times 2^{\lfloor\log_3\frac{n}{i}\rfloor}&=\sum_{i=1}^{n}[2\nmid i]i\times(1+\sum_{j=1}[i\times 3^j\le n]2^{j-1})\\&=\sum_{i=1}^{n}[2\nmid i]i+\sum_{j=1}2^{j-1}\sum_{i=1}^{n}[2\nmid i][i\times 3^j\le n]i\\&=\sum_{i=1}^{n}[2\nmid i]i+\sum_{j=1}2^{j-1}\sum_{i=1}^{\lfloor\frac{n}{3^j}\rfloor}[2\nmid i]i.\end{aligned}$$

通过等差数列求和容易得到, $\sum_{i=1}^{n}[2\nmid i]=\lceil\frac{n}{2}\rceil^2$, 故上式进一步化为

$$\sum_{i=1}^{n}[2\nmid i]i+\sum_{j=1}2^{j-1}\sum_{i=1}^{\lfloor\frac{n}{3^j}\rfloor}[2\nmid i]i=\lceil\frac{n}{2}\rceil^2+\sum_{j=1}2^{j-1}\lceil\frac{\lfloor\frac{n}{3^j}\rfloor}{2}\rceil^2.$$

左边的一项可以 $O(1)$ 计算, 右边的一项显然当 $j\ge O(\log n)$ 后为 0, 故可以 $O(\log n)$ 计算.

故对于单组数据, 时间复杂度为 $O(\log n)$, 空间复杂度为 $O(1)$.

致谢

这个问题暂未找到参考资料.

感谢王翔羚同学与我讨论本题做法并给予启发.