CCPC Henan Provincial Contest 2022 Solution

October 2, 2022

Mocha 上小班啦

- Difficulty: Check-in
- Accepted: 388 / 502

- Idea: Bazoka13
- Preparation: Bazoka13

题目大意

- 求有 n 位且每位数字都不同的最小正整数
- $1 \le n \le 20$

题解

显然只有 $1 \le n \le 10$ 时答案不为 -1，且答案为数字串 1023456789 的长度为 n 的前缀。

Hash

- Difficulty: Medium-Hard
- Accepted: 13 / 167

- Idea: Toxel
- Preparation: Bellalabella

题目大意

- 给定一个仅包含 aehn 的循环串 T 和一个 Hash 函数

$$f(S) = \left(\sum_{i=1}^{|S|} g(S_i) \times 31^{|S|-i} \right) \bmod 998\,244\,353$$

其中 $g(\texttt{a}) = 1, g(\texttt{e}) = 2, g(\texttt{h}) = 3, g(\texttt{n}) = 4$

- 问将循环串分成若干子串后，所有子串 Hash 值的和最大是多少
- $1 \le |T| \le 2 \times 10^5$

题解

考虑一个暴力的做法。我们只考虑长度不超过 D 的所有子串，用 DP 来计算所有分割的答案。由于原串是一个环，我们可以枚举前 D 个位置作为串的起点。考虑 DP 的状态转移有

$$dp_i = \max_{\max(0,i-D)\le j<i}\{dp_j + f(S[j+1,i])\}$$

时间复杂度是 $\mathcal{O}(D^2 n)$。接下来证明 D 取 15 即可。

题解 cont'd

我们只需要证明所有长度不小于 16 的字符串都不是最优分割的一个子串。

设 $P = 998\,244\,353$，考虑长度为 7 的任意字符串 S，注意到

$$31^6 \equiv -110\,740\,672 \pmod{P}$$

$$4 \times 31^6 \equiv 555\,281\,665 \pmod{P}$$

并且有

$$5 \times 31^5 \equiv 143\,145\,755 \pmod{P}$$

可知只要 S 的首字母为 e,n,h 之一，即有

$$\frac{P}{2} \leq f(S) < P$$

题解 cont'd

接下来考察 S 首字母为 a 的情况。注意到

$$31^6 \equiv -110\,740\,672 \pmod{P}$$

$$3 \times 31^5 \equiv 85\,887\,453 \pmod{P}$$

即知 S 前缀为 aa 或 ae 时也有

$$\frac{P}{2} \le f(S) < P$$

综上所述，对于任意长度为 8 的字符串 T，存在 T 的子串 T' 使得

$$\frac{P}{2} \le f(T') < P$$

题解 cont'd

对于任意长度不小于 16 的字符串 T_0, 其前半段存在 $f(T_1) \geq P/2$ 的子串 T_1，后半段存在 $f(T_2) \geq P/2$ 的子串 T_2，从而有

$$f(T_0) < P \leq f(T_1) + f(T_2)$$

即知将 T_0 划分为两段更优，于是 T_0 不是最优分割的一个子串。

Serval 的试卷答案

- Difficulty: Medium-Hard
- Accepted: 7 / 91

- Idea: Serval
- Preparation: Serval

题目大意

- 给定一个仅包含 ABCD 的长度为 n 的字符串 S
- 一张试卷包含若干道不定项选择题
- 一道不定项选择题的答案是 ABCD 的非空子序列
- 所有不定项选择题的答案依次连接得到试卷答案
- 给定 q 次操作，操作分为以下两种：
  - 给定 l, r，对于 $l \le x \le r$，循环移动选项 S_x (A→B，B→C，C→D，D→A)
  - 给定 l, r, k，问有多少恰好 k 道题的试卷答案为 $S[l, r]$ 答案对 998244353 取模
- $1 \le n \le 10^5$，$1 \le q \le 10^5$

题解

考虑查询操作如何求出答案。

类似隔板法，考虑如何放置隔板将 $S[l,r]$ 划分为 k 道不定项选择题。

考虑 $S[l,r]$ 中相邻的两个字符，如果 $S_x \geq S_{x+1}$，那么 S_x 与 S_{x+1} 不可能为同一道题的答案，因此二者之间必须放置隔板。不妨将这些隔板的数量记作 c。

剩下的可以放置隔板的位置可放可不放，于是答案即为：

$$\binom{r-l-c}{k-1-c}$$

题解 cont'd

线段树维护 S 相邻字符的大小关系，查询区间有多少对相邻字符满足前者大于后者即可求得 c。组合数可以通过预处理阶乘及其逆元计算得到。

对于修改操作，可以直接在线段树上区间修改，记录每个结点所在的区间当前循环移动的次数。

时间复杂度 $\mathcal{O}(16n\log n)$ 或 $\mathcal{O}(4n\log n)$。

分块维护 S 视常数而定也可通过。

Mocha 上中班啦

- Difficulty: Hard
- Accepted: 0 / 8

- Idea: Bazoka13
- Preparation: Bazoka13

题目大意

- 给定一个 n 个点的凸包
- 凸包会绕着给定的旋转中心以 $1^\circ/\mathrm{s}$ 的速度旋转
- 询问旋转一周的过程中凸包上的所有点均严格位于两条平行线内部的总时间
- $3 \leq n \leq 10^5$，坐标范围 $[-10^9, 10^9]$

题解

我们定义当凸多边形上所有的点都**严格**位于两条直线之间（即位于两条直线之间且不在任意一条直线上）时，凸多边形处于合法状态，否则处于非法状态。那么题意即为在凸多边形旋转一周的过程中有多长时间处于合法状态。可以发现，当凸多边形处于非法状态时，一定至少有一个端点在直线上或者在两条直线外部的区域。我们对于每一个端点，求出其在直线上或者在两条直线外部的区域的时间区间。答案即为 $360-$ 所有端点时间区间的并集的长度。

Serval 的俳句

- Difficulty: Easy
- Accepted: 344 / 834

- Idea: Serval
- Preparation: Serval

题目大意

- 给定一个仅包含小写英文字母的字符串 S
- 从中找到一个满足下述条件子序列 S' 或判断其不存在:
  - $S'_1, \ldots, S'_5$ 为同一个字符
  - $S'_6, \ldots, S'_{12}$ 为同一个字符
  - $S'_{13}, \ldots, S'_{17}$ 为同一个字符
- $|S| \leq 10^6$

解法一

粗略地估计，$|S| \geq 26 \times 16 + 1 = 417$ 时，必然有一个字母出现超过 17 次，此时必定有解。我们只需要考虑 S 的长度至多为 417 的前缀即可。事实上，这个界可以更小。

枚举 S' 的三段字符 c_1, c_2, c_3，判断 $c_1 \ldots c_2 \ldots c_3 \ldots$ 是否能作为前缀的子串。如果能则 $c_1 \ldots c_2 \ldots c_3 \ldots$ 可以作为答案，如果都不能则答案不存在。

时间复杂度 $\mathcal{O}(|S| + 26^4 \times 17)$。

解法二

先找到 S 的最短的出现 5 次相同字符的前缀，删去这个前缀，并记出现 5 次的字符为 c_1，记剩余的字符串为 S_1。

继续找到 S_1 的最短的出现 7 次相同字符的前缀，删去这个前缀，并记出现 7 次的字符为 c_2，记剩余的字符串为 S_2。

继续找到 S_2 的最短的出现 5 次相同字符的前缀，并记出现 5 次的字符为 c_3。

如果 c_1, c_2, c_3 都能找到，则 $c_1 \ldots c_2 \ldots c_3 \ldots$ 可以作为答案。如果找不到，则答案不存在。

时间复杂度 $\mathcal{O}(|S|)$。

集合之和

- Difficulty: Easy
- Accepted: 295 / 1024

- Idea: Serval
- Preparation: Serval

题目大意

- 定义有限数集 A, B 的加法为

$$A + B = \{x + y \mid x \in A, y \in B\}$$

- 给定 n，构造数集 A 满足下列条件:
  - $|A + A| = n$
  - $\forall x \in A : x \in \mathbb{Z} \cap [0, 5 \times 10^5]$
- $1 \le n \le 5 \times 10^5$

题解

先注意到取 $A=\{0,1,2,\ldots,k\}$ 时，$A+A=\{0,1,2,\ldots,2k\}$，有 $|A+A|=2k+1$，这给出了 n 为奇数时的构造。

题解 cont'd

接下来我们先证明 $n=2$ 与 $n=4$ 时无解。
只需要注意到

$$2|A|-1 \leq |A+A| \leq \frac{|A|(|A|+1)}{2}$$

即知 $|A|=1$ 时 $|A+A|=1$，$|A|=2$ 时 $|A+A|=3$，$|A|=3$ 时 $5 \leq |A+A| \leq 6$，因此 $n=2$ 与 $n=4$ 时无解。

题解 cont'd

考虑 n 为偶数时如何构造。一个有趣的想法是从 n 为奇数时构造的 A 中移除 1，这样 $A+A$ 中恰好只被移除了一个 1。也即取 $A=\{0,2,3,\ldots,k\}$ 时，$A+A=\{0,2,3,\ldots,2k\}$，有 $|A+A|=2k$，这给出了 n 为偶数时的构造。

Mocha 上大班啦

- Difficulty: Easy
- Accepted: 244 / 642

- Idea: Bazoka13
- Preparation: Bazoka13

题目大意

- 给定 n 个 01 串以及 q 次操作
- 每次操作会把第 i 个串的 $l-r$ 位替换为和第 j 个串的 $l-r$ 位分别进行与运算的结果
- 第 i 次操作有概率 p_i 成功
- 询问最后 n 个串与运算得到的串中 1 的个数
- $2 \le n \le 1000$，$1 \le m \le 4000$，$1 \le q \le 2 \times 10^5$

题解

考虑所有 01 串的第 i 位，这一位会对答案产生贡献当且仅当所有 01 串的第 i 位均为 1。回到我们的操作，如果所有 01 串的第 i 位均为 1，由于我们进行的是与操作，显然对操作多少次都不会影响任何一个 01 串第 i 位的值。因此我们的操作实际上不会对答案产生影响，直接计算初始 01 串与操作后的 1 的个数即可。

旋转水管

- Difficulty: Medium
- Accepted: 61 / 1100

- Idea: Toxel & Bellalabella
- Preparation: Bellalabella

题目大意

- 给定 $2 \times m$ 的水管图
- 水管可以为 L 和 I 两个其中一种
- 水管可以旋转
- 问从是否存在一个水管的放置方式使得水可以从入口流入、从出口流出
- 多组数据，$1 \leq m \leq 10^5$，$\sum m \leq 5 \times 10^5$

题解

考虑暴力递归模拟水的流动情况：

对于起始时刻，如果下方直接为 L 形水管，则有两种情况，可以认为从这个水管向左流动或者向右流动，否则可以认为从这个水管向下流动。

题解 cont’d

我们会发现对于 L 形水管，如果我们从左或右侧流入，则流出方向只能为上或下，由于我们水管只有两行，因此一定有一个方向流动一次后就会跑出我们的水管或者到达终点，而另一个方向可以继续进行递归模拟。

如果我们从上或下流入，则流出方向为左或右，会发现在之后的递归过程中，向左流动的水不可流到此位置右侧，向右流动的水不可流到此位置左侧。

题解 cont'd

对于 I 形水管水流只能保持原来的流动方向继续流动。

考虑完上述情况后会发现每个水管最多被访问过 2 次，因此单组询问的复杂度为 $\mathcal{O}(m)$，总的复杂度为 $\mathcal{O}(\sum m)$。

在模拟时需要记录哪些位置被访问过，否则会死循环。

Oshwiciqwq 的电梯

- Difficulty: Medium
- Accepted: 18 / 66

- Idea: Oshwiciqwq
- Preparation: Oshwiciqwq

题目大意

- 模拟题，略。

题解

最暴力的方法是直接模拟。由于时间范围不大，每一秒模拟所有电梯和人的动作即可。

也有比较简单的方法。在题目给定的电梯运行策略下，对于某一秒和某一电梯可以 $\mathcal{O}(1)$ 算出该电梯此时在哪个坐标，也可以算出电梯接下来到某个坐标的时间。而一个人搭乘的电梯也是固定的，因此可以对于每个人计算出依次进入、走出每个电梯的时间，最后对所有事件排序输出。

Mex Tree

- Difficulty: Medium
- Accepted: 68 / 478

- Idea: JJLeo
- Preparation: Bellalabella

题目大意

- 给定一棵树，每个点的权值为 $0 \sim n-1$ 的一个排列
- 问对于所有 $0 \le k \le n$ 求最大的联通块大小使得联通块权值的 $\mathrm{mex} = k$
- $1 \le n \le 10^6$

题解

设我们此时要计算 $\mathrm{mex} = k$ 时的答案。

当 $k = n$ 时，答案显然为 n，因为可以选择整棵树，此时答案即为 n。

当 $k \neq n$ 时，将 $v_x = 0$ 的点 x 作为整棵树的根。

当 $k = 0$ 时，选择的非空连通子图不能包含权值 0，答案即为根节点所连的子树大小的最大值。

题解 cont'd

当 $k = i$ $(i = 1, 2 \ldots, n-1)$ 时，设 $v_y = k$，我们选择的非空连通子图至少要包含根节点 x 并且点 y 不能在选择的非空连通子图中，因此点 y 子树中的所有节点也不能在选择的非空连通子图中。

若点 y 的子树中包含点 u 其中 $v_u < k$，那么答案即为 0，否则容易发现答案即为 $n - size_y$，其中 $size_y$ 为点 y 的子树大小。

因此我们对于每个点维护子树最小值 mn_y，如果 $mn_y > v_y$ 则 $k = v_y$ 的答案即为 $n - size_y$，否则为 0。

总复杂度为 $\mathcal{O}(n)$。

复合函数

- Difficulty: Medium-Hard
- Accepted: 3 / 124

- Idea: Serval
- Preparation: Serval

题目大意

- 给定正整数 n，并记 $I_n = \{1, 2, \ldots, n\}$
- 给定定义在 I_n 上的函数 f，且对于任意 $x \in I_n$ 有 $f(x) \in I_n$
- 对于任意正整数 k，定义 $f^k(x)$ 如下：

$$f^k(x) = \begin{cases} f(f^{k-1}(x)) & , k > 1 \\ f(x) & , k = 1 \end{cases}$$

- 给出 q 组询问，每组询问给定两个正整数 a, b，求满足 $f^a(x) = f^b(x)$ 的 x 的个数
- $1 \leq n \leq 10^5$，$1 \leq q \leq 10^5$，$1 \leq a, b \leq 10^{18}$

题解

考虑一张 n 个点 n 条边的有向图 G，其中点 x 向 $f(x)$ 连边。容易发现 G 由若干个基环树构成。

对于一组询问，基环树之间是独立的，接下来我们只考虑一棵基环树对答案的贡献。

记基环树的环长为 c，对于基环树内的点 x，$f^a(x) = f^b(x)$ 成立当且仅当以下两个条件至少一个成立：

- $a = b$
- $c \mid |a - b|$ 且 $f^{\min(a,b)}(x)$ 在环上

题解 cont’d

其中 $a = b$ 是平凡的，特判即可。只需要考虑如何求第二类情况的答案。

注意到条件中只需要关注环长以及基环树上各点的深度，因此可以将环长相同的基环树的信息合并处理。可以发现，G 中基环树的环长最多只有 $\mathcal{O}(\sqrt{n})$ 种。对于每种环长，统计深度不超过 $d = 0, 1, 2, \ldots$ 的点的数量，从而可以枚举环长回答询问。

时间复杂度 $\mathcal{O}(n\sqrt{n})$。

串串串串……

- Difficulty: Very Hard
- Accepted: 0 / 6

- Idea: Toxel
- Preparation: JJLeo

题目大意

- 给定 n 个字符串 $s_1, s_2, \ldots, s_n$，定义 $f(t)$ 为字符串 t 的最短合法划分长度，一个划分是合法的当且仅当每个字符串至少是 $s_1, s_2, \ldots, s_n$ 中 m 个字符串的子串。特别地，如果对于字符串 t 不存在合法划分，则 $f(t) = 0$。设字符集为 $1, 2, \ldots, 10^9$、长度在 $[l, r]$ 的字符串的集合为 S，求 $\left(\sum_{t\in S} f(t)\right) \bmod 998\,244\,353$。
- $1 \le m \le n \le 5000$，$1 \le l \le r \le 10^{18}$，$\sum_{i=1}^{n} |s_i| \le 5000$

题解

首先，对 $s_1, s_2, \ldots, s_n$ 建广义后缀自动机，并将状态抽象为节点、转移抽象为有向边。接着将出现次数 $\geq m$ 的节点的导出子图对应的有限状态自动机设为 M，并定义其接受的字符串集合为 $L(M)$。如果 t 中存在一个字符不在 $L(M)$ 中，则 $f(t)$ 恒为 0，下面不考虑这种情况。

可以证明，对于字符串 t，一定存在一个最短合法划分 $(T_1, T_2, \ldots, T_{f(t)})$ 满足 $G(T_i, F(T_{i+1})) \notin L(M)$ 对 $1 \leq i < f(t)$ 成立，其中 $F(s)$ 表示字符串 s 的第一个字符所组成的字符串，$G(s, t)$ 表示将字符串 s 和 t 拼接。

题解 cont'd

具体来说，可以将任意一个最短合法划分 $(T_1, T_2, \ldots, T_{f(t)})$ 调整至满足上述条件：如果 $f(t) = 1$ 显然已经满足条件；否则，若 $G(T_1, F(T_2)) \in L(M)$，可以将 T_1 变为 $G(T_1, F(T_2))$ 并删掉 T_2 的第一个字符，重复该操作直到 $G(T_1, F(T_2)) \notin L(M)$；上述过程中 T_2 一定不为空，否则就得到了一个更短的合法划分，矛盾。我们依次对 $T_2, T_3, \ldots, T_{f(t)-1}$ 进行上述调整即可得到满足条件的最短合法划分。

由上述结论，我们得到了一个利用 M 计算 $f(t)$ 的方法，在 M 上从起始点开始依次匹配 t 的每个字符，如果出现失配就回到起始点重新匹配，$f(t)$ 即为失配次数 $+1$。

题解 cont'd

考虑如下的动态规划，设 $dp_{i,j}$ 表示长度为 i 、按上述方法在 M 上匹配后到达节点 j 的字符串数量，V 和 E 分别是 M 的点集和边集，节点分别用 $1, 2, \ldots, |V|$ 来表示，M 的起始点为 1，(u, v) 表示从 u 指向 v 的一条边，$c(u, v)$ 表示该边对应的字符，$[P]$ 表示艾佛森括号，则有如下转移：

$$dp_{i,j} = \begin{cases} [j = 1], & i = 0 \\ \displaystyle\sum_{(k,j)\in E} dp_{i-1,k}, & i > 0 \land (1, j) \notin E \\ \displaystyle\sum_{(k,j)\in E} dp_{i-1,k} + \sum_{\substack{k\in V \\ \forall (k,l)\in E, c(k,l)\neq c(1,j)}} dp_{i-1,k}, & i > 0 \land (1, j) \in E \end{cases}$$

题解 cont'd

设 $ans_i = \sum\limits_{|t|=i} f(t)$ ，C 为 M 中出现的字符种类数，则有：

$$ans_i = \begin{cases} 1, & i = 0 \\ C \cdot ans_{i-1} + \sum\limits_{j \in V} [(1, j) \in E] dp_{i,j}, & i > 0 \end{cases}$$

设 $N = \sum_{i=1}^{n} |s_i|$，由此我们得到了一个时间复杂度为 $\mathcal{O}(Nr)$ 的算法。

题解 cont'd

注意到当 $i > 0$ 时，设 $A_i = \sum_{j=0}^{i} ans_j$，可知

$$\{dp_{i,1},\ dp_{i,2},\ \ldots,\ dp_{i,|V|},\ ans_i,\ A_i\}$$

中的任意元素可由

$$\{dp_{i-1,1},\ dp_{i-1,2},\ \ldots,\ dp_{i-1,|V|},\ ans_{i-1},\ A_{i-1}\}$$

线性表出，因此使用矩阵快速幂就可以得到一个 $\mathcal{O}\left(N^3 \log r\right)$ 的算法。

题解 cont'd

进一步，设 $a_i = (dp_{i,1}, dp_{i,2}, \dots, dp_{i,|V|}, ans_i, A_i)^\top$，当 $i > 0$ 时有 $a_i = Ba_{i-1}$，其中 B 是一个 $(|V|+2) \times (|V|+2)$ 的方阵。记 $k = |V| + 2$，设 $\varphi(\lambda) = \sum_{i=0}^{k} b_i\lambda^i$ 为 B 的特征多项式，由 Cayley-Hamilton 定理可得 $\varphi(B) = O$，因此有：

$$b_k B^k = -\sum_{j=0}^{k-1} b_j B^j$$

$$b_k a_{i+k} = b_k B^k a_i = -\sum_{j=0}^{k-1} b_j B^j a_i = -\sum_{j=0}^{k-1} b_j a_{i+j}$$

题解 cont'd

因为向量 a_i 的每一维独立，所以有：

$$b_k A_{i+k} = -\sum_{j=0}^{k-1} b_j A_{i+j}$$

从而序列 $A_0, A_1, A_2, \ldots$ 满足 k 阶常系数线性齐次递推，可以使用上述动态规划求解出前 $\mathcal{O}(N)$ 项 A_i，再使用 Berlekamp-Massey 算法在 $\mathcal{O}(N^2)$ 的时间内求出最短递推式，最后使用多项式相关科技（参见此处）在 $\mathcal{O}(N \log N \log r)$ 的时间复杂度求解 A_{l-1} 和 A_r。

总时间复杂度为 $\mathcal{O}(N^2 + N \log N \log r)$。