洛谷蓝桥杯模拟赛 & 洛谷算法基础赛 #3 讲评

shinzanmono

2025 年 3 月 29 日

A. 在小小的日历里面数呀数呀数

今天是 2025 年 3 月 29 日，现在我们想知道距离下一个完全平方年的 1 月 1 日还有几天（即从 2025 年 3 月 29 日 0 时到下次完全平方年的 1 月 1 日 0 时）。

A. 在小小的日历里面数呀数呀数

下一个平方年为 2116 年。
首先计算从今天到 2025 年 12 月 31 日的天数，为 278 天。
我们计算 2026 年 1 月 1 日到 2115 年 12 月 31 日的天数。其中包含 2028, 2032, ..., 2096, 2104, 2108, 2112 共 21 个闰年，共 $365 \times 90 + 21 = 32871$ 天。
将两个值加起来，共 33149 天，答案即为 33149。
另解：在浏览器控制台输入 `(Date.parse("2116-01-01 00:00")-Date.parse("2025-03-29 00:00"))/1000/60/60/24` 即可：

B. 我是黄色恐龙大将军

对于正整数 n，设 a_n 为 2^n 在十进制下的最高非零位的值，b_n 为 5^n 在十进制下的最高非零位的值，求所有可能的作为 $a_n \times b_n$ 的值的和。

B. 我是黄色恐龙大将军

一个显然的想法是直接打表找规律，发现 $n \le 20$ 的时候答案为 45。然后猜测最终答案也为 45。

考虑证明。设 s_n 为 2^n 的位数，t_n 为 5^n 的位数，则必有 $a_n 10^{s_n-1} \le 2^n < (a^n+1)10^{s_n-1}, b_n 10^{t_n-1} \le 5^n < (b^n+1)10^{t_n-1}$，两式相乘可得 $a_n b_n \le 10^{n-(s_n+t_n-2)} < (a_n+1)(b_n+1)$。而 $n \ne 0$，所以 $n-(s_n+t_n-2)=1$，代入可得：

$$\begin{cases} a_n b_n \le 10 \\ 10 < (a_n+1)(b_n+1) = a_n b_n + a_n + b_n + 1 \le 2a_n b_n + 1 \end{cases}$$

即 $5 \le a_n b_n \le 10$，取 $n = 9, 4, 7, 3, 5, 1$ 可取到全部值。

C. 化食欲为动力

给定长度分别为 n, m, k 的三个数组 a，b，c，对于 $i \le n, j \le m, t \le k$，求 $a_i \times b_j \bmod c_t$ 的最大值。
$n, m, k \le 200$，$a_i, b_j, c_t \le 10^9$

C. 化食欲为动力

首先考虑数据范围支持使用 $O(nmk)$ 的枚举来解决问题。
考虑 $a_i \times b_j$ 的最大值可能达到 10^{18}，所以要用 64 位整数，即 C++ 中的 long long。

D. 哇，这就是 5p

有一个序列 a，第 i 个数有 p_i 的概率对结果有 a_i 的贡献，求贡献是 m 的倍数的概率。
$1 \le n \le 10^5$，$1 \le m \le 1000$。

D. 哇，这就是 5p

对于 $n \le 15$ 的部分，二进制枚举即可。
对于 $n \le 1000$ 的部分，考虑 $f_{i,j}$ 表示前 i 个数和为 j 的概率，由于我们只关心模 m 意义下的值，所以可以直接将第二维对 m 取模，转移为 $f_{i,j} \leftarrow (1-p_i) \cdot f_{i-1,j} + p_i \cdot f_{i-1,(j-a_i) \bmod m}$，时空复杂度都为 $O(nm)$。
我们发现 f_i 的值仅依赖于 f_{i-1}，所以我们可以对第一维使用滚动数组进行优化，空间复杂度降为 $O(m)$。

注意不寻常的模数。

E. 投入严厉地本地

在本题中，对于一个字符串 x，我们定义：

- $|x|$ 表示 x 的长度，空字符串长度为 0。
- $x_{i\sim j}$ 表示 x 的第 i 个字符到第 j 个字符按顺序连接以后形成的子串，例如 $\texttt{abcd}_{2\sim 4}=\texttt{bcd}$。
- y 是 x 的前缀当且仅当存在一个 p 满足 $x_{1\sim p}=y$。
- y 是 x 的后缀当且仅当存在一个 p 满足 $x_{p\sim |x|}=y$。

字符串的字符集是小写字母集合，即字符串仅由小写字母构成。

E. 投入严厉地本地

给定两个字符串 s, t，和一个参数 k。此外有一个映射规则集合 $f = \{(\lambda_i, \gamma_i) | i = 1, 2, 3, \dots m\}$。其中 λ_i 是长度为 k 的字符串，γ_i 是一个长度为 1 的字符串，或一个空字符串，λ_i 互不相同，m 是映射规则的数量。

已知对于映射规则集合 f，s 可以按如下流程生成字符串 t：

1. 令 $i = 1$。
2. 如果 $i > |s|$，生成结束。
3. 如果存在一个 $j \in [1, m]$ 使得 λ_j 是 $s_{1\sim i}$ 的后缀，则令 $t := t \circ \gamma_i$，这里 $:=$ 表示赋值，$\circ$ 表示字符串拼接。
4. 如果对任何的 $j \in [1, m]$ 都有 λ_j 不是 $s_{1\sim i}$ 的后缀，则令 $t := t \circ s_{i\sim i}$。
5. 令 $i := i + 1$，返回 2。

现在，给定 s 和由它生成的字符串 t，以及参数 k，你需要给出一个映射规则集合 f，使得 s 按映射规则 f 生成的字符串是 t。

E. 投入严厉地本地

考虑到数据范围很小，而有用字符串的只有 s 的每个前缀的长度为 k 的后缀。我们将这些后缀提取出来，然后通过搜索枚举每个后缀匹配哪些字符，并暴力判断即可。复杂度为 $O((|s|-k)^{|t|}|s||t|\log m)$。

F. 扶苏出勤日记

你有两个长度为 n 的序列 a, b，和两个变量 s, c，初始时 $s = c = 0$，你需要找到满足如下操作过程中 s 恒为非负的 d 的最大值。
依次遍历 (a_i, b_i) 进行如下操作。

1. $c \leftarrow c + b_i$
2. 选择一个整数 p 满足 $0 \leq p \leq c$，令 $c \leftarrow c - p$，$s \leftarrow s + p \times a_i$。
3. $s \leftarrow s - d$

F. 扶苏出勤日记

显然答案的存在具有单调性，可以二分答案，考虑如何 check。
显然可以想到贪心，对于第 i 个位置，我们不断找到最小的 i 满足 $b_i \neq 0$ 并尽可能将 s 买到恰好不小于 d 就停止。
这个贪心是对的，我们有结论：当 r 固定时，$\max_{i=l}^{r} a_i$ 单调不升。所以我们买最左面的一定是最优决策。
我们需要找到一个方法静态查询区间 a_i 的最大值。可以想到使用单调队列，具体地，维护单调递减的队列，每次扫到一个点 i 的时候，其一定会在队首那天购买，如果他自己就是队首，那么买完以后将其弹出队列即可，这样可以做到复杂度为 $O(n \log V)$。
也可以使用 st 表，这样总复杂度为 $O(n(\log n + \log V))$。

G. 在小小的奶龙山里面挖呀挖呀挖

给定一棵 n 个节点的树，q 次询问，每次求路径上点的乘积的本质不同的质因子个数。

$1 \le n, q \le 5 \times 10^4$，$1 \le V \le 10^5$。

G. 在小小的奶龙山里面挖呀挖呀挖

记 $\pi(n)$ 表示 $1 \sim n$ 中的素数个数，点 u 的父亲为 fa_u，u 和 v 的最近公共祖先记为 $\text{lca}(u,v)$。设树根为 1，n,q 同阶。
对于 $n \le 100$ 的部分，直接暴力 $O(qn\pi(\sqrt{V}))$ 求解即可。
对于 $n \le 1000$ 的部分，我们发现 $\pi(V)$ 较小，只有大概 10^4 左右，所以设 $f_{u,p}$ 表示 $1 \sim u$ 路径上质因子 p 的出现次数，而 u 到 v 路径上的质因子 p 出现次数为
$f_{u,p} + f_{v,p} - f_{\text{lca}(u,v),p} - f_{fa_{\text{lca}(u,v)},p}$。时空复杂度均为 $O(n\pi(V))$。但是其实可以时间多带一个 $\log n$ 也没有关系。
对于 $n \le 50000$ 的部分，我们开不下二维数组，所以考虑将询问离线，对每个质数开一个数组即可通过。此题需要一些基本的卡常技巧，可以将树拍到 DFS 序上，每次遍历时直接迭代搜索，并且将询问离线的时候记录 LCA。

G. 在小小的奶龙山里面挖呀挖呀挖

Bonus：$n, q \le 3 \times 10^5$，$V \le 10^8$ 怎么做。
首先考虑 $> \sqrt{V}$ 的质因子一个数最多出现一个。所以我们可以对于 $> \sqrt{V}$ 的质因子进行树上莫队，时间复杂度 $O(n\sqrt{n})$。
对于 $\le \sqrt{V}$ 的质因子，我们可以使用较小常数的 $O\left(\frac{n\pi(\sqrt{V})}{w}\right)$ 的树上静态链查询即可。
需要查询每个数的最大质因子，这个根据实现方法不同可以 $O(n\omega(V))$ 或 $O(n\log V)$ 解决。

H. 吃猫粮的玉桂狗

给定一棵以 1 为根的有根树，有 m 种颜色，t 个限制 (a_i, b_i)，每种颜色有出现次数上限 c_i。你需要对每一个结点染一种颜色，设第 i 个结点的颜色为 col_i。定义合法的染色方式为：第 i 种颜色出现次数 $\le c_i$ 且对于每个限制 (a_k, b_k)，你要保证不存在一对父子结点 (fa_u, u) 满足 $col_{fa_u} = a_k$ 且 $col_u = b_k$。

$n, m \le 50$，$\left\lfloor \frac{n}{2} \right\rfloor \le c_i \le n$。

H. 吃猫粮的玉桂狗

对于 (a_i, b_i) 的限制，是比较容易处理的，记 $f_{u,i}$ 表示点 u 染为颜色 i 个方案数，直接树形 dp 计算就好了。
而这样会导致可能有颜色超出了限制，考虑 c_i 至少为 n 的一半，则至多会有一种颜色超出限制，我们枚举这种颜色 x，并计算 x 的出现次数超过 c_x 的方案数，从答案中减掉即可。计算可以使用树上背包，注意树上背包的写法，在搜索 u 的子节点 v 的时候要先转移再更新 u 的子树大小。
时间复杂度 $O(n^2m^3)$。

注意不寻常的模数。