题目选讲：动态规划 & 数论

清华大学　任舍予

2024 年 8 月 9 日

问题 (NOIP 2017 提高组小凯的疑惑)

给定 a, b，求最大的不能被 a, b 的非负整数倍之和表示的正整数。
数据范围：$1 \leq a, b \leq 10^9$。

- 根据裴蜀定理，对于任意正整数 n，存在 x_0, y_0 使得 $n = ax + by$ 的所有整数解恰为 $x = x_0 + kb, y = y_0 - ka$。

- 根据裴蜀定理，对于任意正整数 n，存在 x_0, y_0 使得 $n = ax + by$ 的所有整数解恰为 $x = x_0 + kb, y = y_0 - ka$。
- x, y 不能同时为非负整数当且仅当 $x \geq 0$ 时一定有 $y < 0$，此时可能取到的最大值为 $x = b - 1, y = -1$。

- 根据裴蜀定理，对于任意正整数 n，存在 x_0, y_0 使得 $n = ax + by$ 的所有整数解恰为 $x = x_0 + kb, y = y_0 - ka$。
- x, y 不能同时为非负整数当且仅当 $x \geq 0$ 时一定有 $y < 0$，此时可能取到的最大值为 $x = b - 1, y = -1$。
- 代入计算得到 $n = a(b-1) + b(-1) = ab - a - b$。

- 根据裴蜀定理，对于任意正整数 n，存在 x_0, y_0 使得 $n = ax + by$ 的所有整数解恰为 $x = x_0 + kb, y = y_0 - ka$。
- x, y 不能同时为非负整数当且仅当 $x \geq 0$ 时一定有 $y < 0$，此时可能取到的最大值为 $x = b - 1, y = -1$。
- 代入计算得到 $n = a(b-1) + b(-1) = ab - a - b$。
- 时间复杂度 $O(1)$，空间复杂度 $O(1)$。

问题 (NOI 2018 屠龙勇士)

有 n 条龙，每条龙的初始生命为 a_i，使用恢复能力时每次恢复 p_i。初始有 m 柄剑。玩家选定一个攻击次数 x。

玩家按照 $1 \sim n$ 的顺序分别屠龙：

- 每次会选择当前拥有的，攻击力不高于巨龙初始生命值中攻击力最大的一把剑作为武器。如果没有这样的剑，则选择攻击力最低的一把剑作为武器。玩家将使用这柄剑攻击 x 次。
- 攻击后，巨龙会不断使用恢复能力。若恢复过程中生命值为 0，巨龙死亡。否则游戏失败。
- 攻击结束后，这柄剑消失，玩家获得一柄新的剑。

求最小的 x，使得能够屠掉所有巨龙。

数据范围：$1 \le n, m \le 10^5$，$\mathrm{lcm}\{p_i\} \le 10^{12}$，$a_i \le p_i$。

- 每次使用的剑是固定的，可以使用数据结构快速求出。

- 每次使用的剑是固定的，可以使用数据结构快速求出。
- 设攻击第 i 条巨龙所选的剑攻击力为 r_i，则需要满足：$p_i \mid a_i - r_i x$。

- 每次使用的剑是固定的，可以使用数据结构快速求出。
- 设攻击第 i 条巨龙所选的剑攻击力为 r_i，则需要满足：$p_i \mid a_i - r_i x$。
- 取 $g = \gcd(r_i, p_i)$，转化为 $\frac{r_i}{g} x \equiv \frac{a_i}{g} \pmod{\frac{p_i}{g}}$。

- 每次使用的剑是固定的，可以使用数据结构快速求出。
- 设攻击第 i 条巨龙所选的剑攻击力为 r_i，则需要满足：$p_i \mid a_i - r_i x$。
- 取 $g = \gcd(r_i, p_i)$，转化为 $\frac{r_i}{g} x \equiv \frac{a_i}{g} \pmod{\frac{p_i}{g}}$。
- 直接乘逆元到另一侧即转化为标准的一元一次同余方程组形式即可合并。

- 每次使用的剑是固定的，可以使用数据结构快速求出。
- 设攻击第 i 条巨龙所选的剑攻击力为 r_i，则需要满足：$p_i \mid a_i - r_i x$。
- 取 $g = \gcd(r_i, p_i)$，转化为 $\frac{r_i}{g} x \equiv \frac{a_i}{g} \pmod{\frac{p_i}{g}}$。
- 直接乘逆元到另一侧即转化为标准的一元一次同余方程组形式即可合并。
- 时间复杂度 $O(n \log m + n \log \operatorname{lcm}\{p_i\})$，空间复杂度 $O(n + m)$。

问题

求 $n!$ 末尾 0 的个数与最后一位非 0 数字。
数据范围： $1 \le n \le 10^{18}$。

- 统计 0 的个数只需要看 5 的幂次，直接计算 $\sum_{k=1}^{+\infty}\lfloor n/5^k\rfloor$。

- 统计 0 的个数只需要看 5 的幂次，直接计算 $\sum_{k=1}^{+\infty}\lfloor n/5^k\rfloor$。
- 设 5 的幂次为 c，最后一位即为 $n!/10^c \bmod 10$，考虑分别计算模 2 和 5 的值。

- 统计 0 的个数只需要看 5 的幂次，直接计算 $\sum_{k=1}^{+\infty}\lfloor n/5^k\rfloor$。
- 设 5 的幂次为 c，最后一位即为 $n!/10^c \bmod 10$，考虑分别计算模 2 和 5 的值。
- 对于 $n \geq 2$，模 2 的值显然为 0。

- 统计 0 的个数只需要看 5 的幂次，直接计算 $\sum_{k=1}^{+\infty} \lfloor n/5^k \rfloor$。
- 设 5 的幂次为 c，最后一位即为 $n!/10^c \bmod 10$，考虑分别计算模 2 和 5 的值。
- 对于 $n \geq 2$，模 2 的值显然为 0。
- 模 5 的值同样可以递归计算：所有 5 的倍数在 $\div 5$ 之后即为 $(n/5)!$。

- 统计 0 的个数只需要看 5 的幂次，直接计算 $\sum_{k=1}^{+\infty} \lfloor n/5^k \rfloor$。
- 设 5 的幂次为 c，最后一位即为 $n!/10^c \bmod 10$，考虑分别计算模 2 和 5 的值。
- 对于 $n \geq 2$，模 2 的值显然为 0。
- 模 5 的值同样可以递归计算：所有 5 的倍数在 $\div 5$ 之后即为 $(n/5)!$。
- 时间复杂度 $O(\log_5 n)$，空间复杂度 $O(1)$。

问题 (CF623B)

给定序列 $[a_1, \ldots, a_n]$，有两种操作：

- 删除一个长度为 m 的区间，代价为 $m \times b$，总共只能操作一次。
- 对于每一个元素，可以将其 $+1$ 或 -1，代价为 c，每个元素只能操作一次。

求让剩下的数的 $\gcd > 1$ 的最小花费。
数据范围：$1 \leq n \leq 10^6$，$2 \leq a_i \leq 10^9$。

- 显然只需要考虑 gcd 为素数的情况。

- 显然只需要考虑 gcd 为素数的情况。
- 由于 a_1, a_n 不可能同时删除，只需要枚举 $a_1 - 1, a_1, a_1 + 1, a_n - 1, a_n, a_n + 1$ 的所有素因数。

- 显然只需要考虑 gcd 为素数的情况。
- 由于 a_1, a_n 不可能同时删除，只需要枚举 $a_1 - 1, a_1, a_1 + 1, a_n - 1, a_n, a_n + 1$ 的所有素因数。
- 枚举 $\gcd = p$ 后，每个数可以选择删除或单点修改，直接线性 DP 即可。

- 显然只需要考虑 $\gcd$ 为素数的情况。
- 由于 a_1, a_n 不可能同时删除，只需要枚举 $a_1 - 1, a_1, a_1 + 1, a_n - 1, a_n, a_n + 1$ 的所有素因数。
- 枚举 $\gcd = p$ 后，每个数可以选择删除或单点修改，直接线性 DP 即可。
- 时间复杂度 $O(n\omega(a))$，空间复杂度 $O(n)$。

问题 (洛谷 P8338)

给定排列 $[p_1,\ldots,p_n]$，定义 $v(p)$ 为最小的正整数 k 使得 $p^k=1$。定义 $f(i,j)$ 如下：

- 若存在 k 使得 $p^k\cdot i=j$，则 $f(i,j)=0$；
- 否则令 q 为交换 p_i,p_j 后得到的排列，$f(i,j)=v(q)$。

求 $\sum_{i=1}^n\sum_{j=1}^n f(i,j)$。
数据范围：$1\le n\le 5\times 10^5$。

- 记环长分别为 $r_1, \ldots, r_m$，实际上要求的是

$$2 \sum_{1 \le i < j \le m} r_i r_j \operatorname{lcm}(r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_{j-1}, r_{j+1}, \ldots, r_n, r_i + r_j).$$

- 记环长分别为 $r_1,\ldots,r_m$，实际上要求的是

$$2\sum_{1\le i<j\le m} r_i r_j \operatorname{lcm}(r_1,\ldots,r_{i-1},r_{i+1},\ldots,r_{j-1},r_{j+1},\ldots,r_n,r_i+r_j).$$

- 注意到不同的环长只有 $O(\sqrt{n})$ 种，因此可以枚举两种环长算贡献。

- 记环长分别为 $r_1, \ldots, r_m$，实际上要求的是

$$2 \sum_{1 \le i < j \le m} r_i r_j \operatorname{lcm}(r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_{j-1}, r_{j+1}, \ldots, r_n, r_i + r_j).$$

- 注意到不同的环长只有 $O(\sqrt{n})$ 种，因此可以枚举两种环长算贡献。
- 最朴素的方式是拆素数幂算贡献，直接对每种素数用数据结构维护出 $r_1 \sim r_m$ 的贡献，每次修改时暴力删除与插入。

- 记环长分别为 $r_1, \ldots, r_m$，实际上要求的是

$$2 \sum_{1 \le i < j \le m} r_i r_j \operatorname{lcm}(r_1, \ldots, r_{i-1}, r_{i+1}, \ldots, r_{j-1}, r_{j+1}, \ldots, r_n, r_i + r_j).$$

- 注意到不同的环长只有 $O(\sqrt{n})$ 种，因此可以枚举两种环长算贡献。
- 最朴素的方式是拆素数幂算贡献，直接对每种素数用数据结构维护出 $r_1 \sim r_m$ 的贡献，每次修改时暴力删除与插入。
- 进一步地，删去 r_i、删去 r_j、加入 $r_i + r_j$ 至多只有一个会产生贡献，于是可以直接将三者的贡献相乘。按照这个方法可以做 P9135。

- 记环长分别为 $r_1, \dots, r_m$，实际上要求的是

$$2 \sum_{1 \le i < j \le m} r_i r_j \operatorname{lcm}(r_1, \dots, r_{i-1}, r_{i+1}, \dots, r_{j-1}, r_{j+1}, \dots, r_n, r_i + r_j).$$

- 注意到不同的环长只有 $O(\sqrt{n})$ 种，因此可以枚举两种环长算贡献。
- 最朴素的方式是拆素数幂算贡献，直接对每种素数用数据结构维护出 $r_1 \sim r_m$ 的贡献，每次修改时暴力删除与插入。
- 进一步地，删去 r_i、删去 r_j、加入 $r_i + r_j$ 至多只有一个会产生贡献，于是可以直接将三者的贡献相乘。按照这个方法可以做 P9135。
- 时间复杂度 $O(n)$，空间复杂度 $O(n)$。

问题 (LOJ 563)

给定 $2m-2$ 个点的二分图 $G(m)$，左侧第 i 个点和右侧第 j 个点有连边当且仅当 $i=j$ 或 $ij \equiv 1 \pmod m$。
设 $f(m)$ 表示 $G(m)$ 的本质不同的最大匹配的个数，求 $f(1) \sim f(n)$。
数据范围：$1 \le n \le 10^7$。

- 只需要求 $x^2 \equiv 1 \pmod{n}$ 的解的个数。

- 只需要求 $x^2 \equiv 1 \pmod{n}$ 的解的个数。
- 考虑拆素数幂，计算 $x^2 \equiv 1 \pmod{p^k}$ 的解的个数。

- 只需要求 $x^2 \equiv 1 \pmod{n}$ 的解的个数。
- 考虑拆素数幂，计算 $x^2 \equiv 1 \pmod{p^k}$ 的解的个数。
  - 若 $p > 2$，由于 $\gcd(x-1, x+1) < p$，一定有 $x \equiv \pm 1 \pmod{p^k}$。

- 只需要求 $x^2 \equiv 1 \pmod{n}$ 的解的个数。
- 考虑拆素数幂，计算 $x^2 \equiv 1 \pmod{p^k}$ 的解的个数。
  - 若 $p > 2$，由于 $\gcd(x-1, x+1) < p$，一定有 $x \equiv \pm 1 \pmod{p^k}$。
  - 若 $p = 2$，由于 $\gcd(X-1, x+1) < 4$，一定有 $x \equiv \pm 1, 2^{k-1} \pm 1 \pmod{2^k}$。

- 只需要求 $x^2 \equiv 1 \pmod n$ 的解的个数。
- 考虑拆素数幂，计算 $x^2 \equiv 1 \pmod{p^k}$ 的解的个数。
  - 若 $p > 2$，由于 $\gcd(x-1, x+1) < p$，一定有 $x \equiv \pm 1 \pmod{p^k}$。
  - 若 $p = 2$，由于 $\gcd(X-1, x+1) < 4$，一定有 $x \equiv \pm 1, 2^{k-1} \pm 1 \pmod{2^k}$。
- 线性筛出答案即可。

- 只需要求 $x^2 \equiv 1 \pmod n$ 的解的个数。
- 考虑拆素数幂，计算 $x^2 \equiv 1 \pmod{p^k}$ 的解的个数。
  - 若 $p > 2$，由于 $\gcd(x-1, x+1) < p$，一定有 $x \equiv \pm 1 \pmod{p^k}$。
  - 若 $p = 2$，由于 $\gcd(X-1, x+1) < 4$，一定有 $x \equiv \pm 1, 2^{k-1} \pm 1 \pmod{2^k}$。
- 线性筛出答案即可。
- 时间复杂度 $O(n)$，空间复杂度 $O(n)$。

问题 (CF1656H)

给定大小为 n 的集合 A 与大小为 m 的集合 B，构造 $S_A \subseteq A$ 与 $S_B \subseteq B$ 满足 $\operatorname{lcm}(S_A) = \operatorname{lcm}(S_B)$。

数据范围：$1 \le n, m \le 10^3$，$1 \le a_i, b_i \le 4 \times 10^{36}$。

- 考虑先令 $S_A = A$，$S_B = B$，然后删去不能选的数，即 $a \nmid \operatorname{lcm}(S_B)$ 的数。

- 考虑先令 $S_A = A$，$S_B = B$，然后删去不能选的数，即 $a \nmid \operatorname{lcm}(S_B)$ 的数。
- 转化为判定是否有

$$\gcd_{i=1}^{m} \frac{a}{\gcd(a, b_i)} > 1.$$

- 考虑先令 $S_A = A$，$S_B = B$，然后删去不能选的数，即 $a \nmid \operatorname{lcm}(S_B)$ 的数。
- 转化为判定是否有

$$\gcd_{i=1}^{m} \frac{a}{\gcd(a, b_i)} > 1.$$

- 这可以直接用线段树进行维护。

- 考虑先令 $S_A = A$，$S_B = B$，然后删去不能选的数，即 $a \nmid \operatorname{lcm}(S_B)$ 的数。
- 转化为判定是否有

$$\gcd_{i=1}^{m} \frac{a}{\gcd(a, b_i)} > 1.$$

- 这可以直接用线段树进行维护。
- 时间复杂度 $O(n^2(\log n + \log a))$，空间复杂度 $O(n^2)$。

问题 (CF1684G)

给定函数 `Euclid(a, b)`：在计算 $\gcd(a, b)$ 时将过程中所有 $a \bmod b$ 的值插入一个可重集中。
给定 n, m 与一个大小为 n 的可重集，要求构造若干对值域在 $[1, m]$ 内的 (a, b)，使得分别执行 `Euclid(a, b)` 后恰好得到给定的可重集。
数据范围：$1 \le n \le 10^3$，$1 \le m \le 10^9$。

- 对于可重集中 $\leq m/3$ 的数 x，可以构造 $(3x, 2x)$。

- 对于可重集中 $\le m/3$ 的数 x，可以构造 $(3x, 2x)$。
- 对于可重集中 $> m/3$ 的数 x，过程一定为

$$(2x+y, x+y) \to (x+y, x) \to (x, y) \to \cdots \to (\gcd(x,y), 0).$$

- 对于可重集中 $\leq m/3$ 的数 x，可以构造 $(3x, 2x)$。
- 对于可重集中 $> m/3$ 的数 x，过程一定为

$$(2x+y, x+y) \to (x+y, x) \to (x, y) \to \cdots \to (\gcd(x,y), 0).$$

- 即，序列中一定包含 $\gcd(x,y)$，于是直接令 $y \mid x$ 总是不劣的。

- 对于可重集中 $\le m/3$ 的数 x，可以构造 $(3x, 2x)$。
- 对于可重集中 $> m/3$ 的数 x，过程一定为

$$(2x+y, x+y) \to (x+y, x) \to (x, y) \to \cdots \to (\gcd(x,y), 0).$$

- 即，序列中一定包含 $\gcd(x,y)$，于是直接令 $y \mid x$ 总是不劣的。
- 问题转化为对每个 $> m/3$ 的数找一个值域满足要求的数匹配。

- 对于可重集中 $\leq m/3$ 的数 x，可以构造 $(3x, 2x)$。
- 对于可重集中 $> m/3$ 的数 x，过程一定为

$$(2x+y, x+y) \to (x+y, x) \to (x, y) \to \cdots \to (\gcd(x,y), 0).$$

- 即，序列中一定包含 $\gcd(x,y)$，于是直接令 $y \mid x$ 总是不劣的。
- 问题转化为对每个 $> m/3$ 的数找一个值域满足要求的数匹配。
- 注意到找到的数一定 $\leq m/3$，因此只需要判断二分图是否有完美匹配即可。

- 对于可重集中 $\le m/3$ 的数 x，可以构造 $(3x, 2x)$。
- 对于可重集中 $> m/3$ 的数 x，过程一定为

$$(2x+y, x+y) \to (x+y, x) \to (x, y) \to \cdots \to (\gcd(x, y), 0).$$

- 即，序列中一定包含 $\gcd(x, y)$，于是直接令 $y \mid x$ 总是不劣的。
- 问题转化为对每个 $> m/3$ 的数找一个值域满足要求的数匹配。
- 注意到找到的数一定 $\le m/3$，因此只需要判断二分图是否有完美匹配即可。
- 时间复杂度 $O(n^2)$，空间复杂度 $O(n^2)$。