数学题

算法 1

枚举，暴力判断，复杂度 $\mathrm{O}(n\log n)$，期望得分 30。

算法 2

分段打表，复杂度 $\mathrm{O}(\sqrt{n\log n})$，期望得分 60。（应该不会有人写这一档吧）

算法 3

显然问题 i 简单的充要条件是 $i-1$ 的个位只能是 $0,1,2$，其余位只能是 $0,1,2,3$。

问题变为了求 $0\sim n-1$ 中满足上述限制的数的数量，这里可以用数位 dp，但是因为限制简单我们可以用另一种更容易的实现方式。

具体的，我们可以从高到低枚举与 n 第一个不同的位 i，小于 n 的限制变为了高于 i 位的与 n 完全相同，低 i 位小于 n 的第 i 位，低于 i 位的没有限制。

由于每一位都独立所以贡献大概是一个 $3\cdot 4^{i-1}\cdot\min(n_i+1,4)$ 装物，需要特判第 0 位。

时间复杂度 $\mathrm{O}(\log n)$，期望得分 100。

听音乐

算法 1

我们把序列复制一遍拼到最后那么显然访问过的点是 $n+1$ 的一个邻域（$n+1$ 即复制前的 1）

可以枚举这个邻域的左右边界 l,r，先花费 $m-l-r-\min(l,r)$ 走遍这个邻域，再向其中插入若干播放操作。

这个过程已经可以用堆来贪心了，先将所有 a_i 加入堆，每次拿出堆顶 w_i，将 w_i 贡献进答案并将 w_i-b_i 加入堆。

时间复杂度 $\mathrm{O}(n^2(n+m)\log n)$，实现优秀或许能拿到 100 分。

算法 2

考虑把算法 1 中的贪心换一种维护方式，因为值域不大所以我们可以把每个 i 的 $a_i,a_i-b_i,a_i-2b_i,\dots$ 加入桶最后桶排不断取出最大值。

时间复杂度 $\mathrm{O}(n^2(m+k))$，可以比较稳地拿到 100 分。

选择美食

算法

有个结论，如果一个数 x 满足 $\varphi(x) \mid x$ ，那么一定满足 $x = 2^a \times 3^b(a \geq 1$, $b \geq 0$, $a, b \in \mathbb{Z})$，特别的，1 是唯一一个例外。

我们可以直接用两个数 a, b 来表示一个满足条件的数。而显然，如果 x, y 满足条件，那么 $x \times y$ 也一定满足条件，且 $a_{x\times y} = a_x + a_y$, $b_{x\times y} = b_x + b_y$

设 $f_{i,j}$ 为表示出 $2^a \times 3^b$ 所需要的最小代价，这个问题就转化为了一个背包问题。同时由于 10^9 中满足条件的数其实非常有限，实际上大约只有 300 个，所以多余的数是完全没用的，这样 n 的范围就降到了 300 左右。

这样时间复杂度为 $\mathrm{O}(n \times \log_2 d \times \log_3 d)$ ，而 $\log_2 10^{205} \approx 700$, $\log_3 10^{205} \approx 400$ ，期望得分 100。

结论证明

众所周知，$\varphi(x) = x\prod_{i=1}^{n}\left(1 - \frac{1}{p_i}\right)$

如果 $\varphi(x) \mid x$ ，那么显然 $\prod_{i=1}^{n}\left(1 - \frac{1}{p_i}\right)$ 一定可以化简为 $\frac{1}{q}$ 的形式。

接下来分类讨论：

- 如果 $x = 1$ ，那么显然符合要求。
- 如果 $x = 2^a$，那么 $\prod_{i=1}^{n}\left(1 - \frac{1}{p_i}\right) = \frac{1}{2}$ ，符合要求。
- 如果 $x = 2^a \times 3^b$，那么 $\prod_{i=1}^{n}\left(1 - \frac{1}{p_i}\right) = \frac{1}{6}$ ，符合要求。
- 如果 $x = 3^b$，那么 $\prod_{i=1}^{n}\left(1 - \frac{1}{p_i}\right) = \frac{2}{3}$ ，不符合要求。
- 对于其他情况，因为 $2, 3$ 后面的质因子都是 $5, 7...$ ，而他们任意两个之间的差都大于 1 ，所以不存在其他合法方案。

小 P 与出题

前言

这道题和前面三个的 gap 有点大，但是我相信部分分能填补这部分 gap（大雾

算法 1

对于满足 $k=3$ 的部分。

考虑 dp，设 $dp_{i,j}$ 表示子树 i 中传上来一条以 j 为端点的链的方案数。

由于 $k=3$，在处理子树 u 的时候，如果没有儿子，那么只能上传一个以 u 为端点的链。如果有一个儿子，那么要么这个儿子传上来的链在 u 处断掉，然后从 u 新起一条链。如果两个儿子，那么可以两个儿子传上来的链匹配，可以两个儿子的链都在 u 断掉然后从 u 新起一条链，还可以一条链断掉另一条传上去。

上传和断掉时好计算贡献的，只需要枚举一遍子树内的点，这部分时间复杂度 $\mathrm{O}(n^2)$，两条链配对可以暴力枚举两个子树内两个点判断并算贡献，类似树上背包分析时间复杂度也是 $\mathrm{O}(n^2)$ 的。

总时间复杂度 $\mathrm{O}(n^2)$，期望得分 45 分。

算法 2

对于满足 $n\le 2000$ 的部分。

由于 $k=5$，所以配对的方式会很多样，但我们依旧可以归类，对于每个传上来的链，可以与另一个链配对，可以在 u 处断掉，也可以继续上传。

由于 k 依旧很小，所以配对的方式也不会很多，设这个方案数为 B，那么通过爆搜可得 $k=5$ 时 $B=10$。

于是可以先预处理出所有儿子两两配对的方案数，以及单独断掉的方案数，然后枚举所有配对方案，如果有一条子树的链传了上去，那么贡献就是这个子树内的所有点的 dp 值乘上其他子树配对以及断裂的方案数的积。从 u 新起一条链同样类似。

预处理单独的断掉显然 $\mathrm{O}(n^2)$，两两配对看起来复杂度很寄，但是冷静分析后其实还是 $\mathrm{O}(n^2)$ 的，最后枚举方案算贡献显然是 $\mathrm{O}(n^2Bk)$ 的。

于是总时间复杂度为 $\mathrm{O}(n^2Bk)$，期望得分 30。

算法 3

改 dp 状态为 $dp_{i,j}$ 表示以 i 为根的子树传上来的链长对 m 取模为 j 的方案数。

这里我们可以和算法 2 一样预处理两两配对和单独断掉的方案数，单独断掉是简单的，而两两配对相对困难，我们能做到的是枚举其中一个端点，然后另一个端点的合法长度就是一个区间。

于是这里可以用线段树维护 dp 数组，同时枚举一个端点可以用 dsu on tree 的 trick 保证复杂度。

但是这里有个问题，我们的边长是会取模的，但线段树很明显做不了这个，所以我们可以考虑把线段树换成平衡树。

使用平衡树维护这个 dp，按 j 的大小顺序排列，这时一条路径顶上添加一条边 w 时，相当于把 $[0,m)$ 拆成 $[0,m-w)+[m-w,m)$ 并合并为 $[m-w,m)+[0,m-w)$，并将 $[0,m-w)$ 第一关键字（即链长）加 w，$[m-w,m)$ 的第一关键字加 $w-m$，这都是平衡树好维护的。

现在需要合并若干子树的 dp 数组（平衡树），仍然继续沿用 dsu on tree，即重儿子的平衡树直接继承，轻儿子的的则枚举一遍加到重儿子上。

时间复杂度为 $\mathrm{O}(nB\log^2 n+nkB\log n)$。实现优秀的话期望得分 100 分。