

荣誉称号 (title) 解题报告

Claris

2018 年 5 月 13 日

荣誉称号 (title.c/cpp/pas)

给定一个序列 $a_1, a_2, \dots, a_n$, 将 a_i 增加 1 的代价为 b_i 。

荣誉称号 (title.c/cpp/pas)

给定一个序列 $a_1, a_2, \dots, a_n$, 将 a_i 增加 1 的代价为 b_i 。

给定 k, m , 求最少的调整代价 , 使得对于任意一个整数 $x(2^k \leq x \leq n)$, $a_x + a_{\lfloor \frac{x}{2} \rfloor} + a_{\lfloor \frac{x}{4} \rfloor} + a_{\lfloor \frac{x}{8} \rfloor} + \dots + a_{\lfloor \frac{x}{2^k} \rfloor}$ 都要是 m 的倍数。

荣誉称号 (title.c/cpp/pas)

给定一个序列 $a_1, a_2, \dots, a_n$, 将 a_i 增加 1 的代价为 b_i 。

给定 k, m , 求最少的调整代价 , 使得对于任意一个整数 $x(2^k \leq x \leq n)$, $a_x + a_{\lfloor \frac{x}{2} \rfloor} + a_{\lfloor \frac{x}{4} \rfloor} + a_{\lfloor \frac{x}{8} \rfloor} + \dots + a_{\lfloor \frac{x}{2^k} \rfloor}$ 都要是 m 的倍数。

- 存在 30% 的数据 , $n \leq 1000, m = 2$ 。

荣誉称号 (title.c/cpp/pas)

给定一个序列 $a_1, a_2, \dots, a_n$, 将 a_i 增加 1 的代价为 b_i 。

给定 k, m , 求最少的调整代价 , 使得对于任意一个整数 $x(2^k \leq x \leq n)$, $a_x + a_{\lfloor \frac{x}{2} \rfloor} + a_{\lfloor \frac{x}{4} \rfloor} + a_{\lfloor \frac{x}{8} \rfloor} + \dots + a_{\lfloor \frac{x}{2^k} \rfloor}$ 都要是 m 的倍数。

- 存在 30% 的数据 , $n \leq 1000, m = 2$ 。
- 存在 70% 的数据 , $n \leq 100000$ 。

荣誉称号 (title.c/cpp/pas)

给定一个序列 $a_1, a_2, \dots, a_n$, 将 a_i 增加 1 的代价为 b_i 。

给定 k, m , 求最少的调整代价 , 使得对于任意一个整数 $x(2^k \leq x \leq n)$, $a_x + a_{\lfloor \frac{x}{2} \rfloor} + a_{\lfloor \frac{x}{4} \rfloor} + a_{\lfloor \frac{x}{8} \rfloor} + \dots + a_{\lfloor \frac{x}{2^k} \rfloor}$ 都要是 m 的倍数。

- 存在 30% 的数据 , $n \leq 1000, m = 2$ 。
- 存在 70% 的数据 , $n \leq 100000$ 。
- 对于 100% 的数据 , $n \leq 10^7, k \leq 10, m \leq 200$ 。

荣誉称号 (title.c/cpp/pas)

给定一个序列 $a_1, a_2, \dots, a_n$, 将 a_i 增加 1 的代价为 b_i 。

给定 k, m , 求最少的调整代价 , 使得对于任意一个整数 $x(2^k \leq x \leq n)$, $a_x + a_{\lfloor \frac{x}{2} \rfloor} + a_{\lfloor \frac{x}{4} \rfloor} + a_{\lfloor \frac{x}{8} \rfloor} + \dots + a_{\lfloor \frac{x}{2^k} \rfloor}$ 都要是 m 的倍数。

- 存在 30% 的数据 , $n \leq 1000, m = 2$ 。
- 存在 70% 的数据 , $n \leq 100000$ 。
- 对于 100% 的数据 , $n \leq 10^7, k \leq 10, m \leq 200$ 。
- Shortest judge solution: 1766 bytes

30 分做法

- 存在 30% 的数据 , $n \leq 1000, m = 2$ 。

30 分做法

- 存在 30% 的数据, $n \leq 1000, m = 2$ 。
- 若将 $\lfloor \frac{x}{2} \rfloor$ 看作 x 的父亲, 则可以得到一棵 n 个点的完全二叉树。

30 分做法

- 存在 30% 的数据, $n \leq 1000, m = 2$ 。
- 若将 $\lfloor \frac{x}{2} \rfloor$ 看作 x 的父亲, 则可以得到一棵 n 个点的完全二叉树。
- 问题转化为对于任意一个点, 往上走 k 条边经过的所有点权值和 $\bmod m = 0$ 。

30 分做法

- 设 $f_{x,S}$ 表示考虑 x 的子树, 要求 x 往上 k 个点的权值 $\bmod 2$ 为 S 状态的最小代价。

30 分做法

- 设 $f_{x,S}$ 表示考虑 x 的子树，要求 x 往上 k 个点的权值 $\bmod 2$ 为 S 状态的最小代价。
- 合并子树的 DP 值后，枚举 x 往上第 k 个点的权值进行转移即可。

30 分做法

- 设 $f_{x,S}$ 表示考虑 x 的子树，要求 x 往上 k 个点的权值 $\bmod 2$ 为 S 状态的最小代价。
- 合并子树的 DP 值后，枚举 x 往上第 k 个点的权值进行转移即可。
- 时间复杂度 $O(n2^k)$ 。

70 分做法

- 存在 70% 的数据, $n \leq 100000$ 。

70 分做法

- 存在 70% 的数据, $n \leq 100000$ 。
- 考虑树退化为序列的情况。

70 分做法

- 存在 70% 的数据, $n \leq 100000$ 。
- 考虑树退化为序列的情况。
- 限制等价于任意长度为 $k + 1$ 的连续区间权值和 $\bmod m = 0$ 。

70 分做法

- 存在 70% 的数据, $n \leq 100000$ 。
- 考虑树退化为序列的情况。
- 限制等价于任意长度为 $k+1$ 的连续区间权值和 $\bmod m = 0$ 。
- 首先根据题意有 $(a_1 + a_2 + \dots + a_{k+1}) \bmod m = 0$ 且 $(a_2 + a_3 + \dots + a_{k+2}) \bmod m = 0$ 。

70 分做法

- 存在 70% 的数据, $n \leq 100000$ 。
- 考虑树退化为序列的情况。
- 限制等价于任意长度为 $k+1$ 的连续区间权值和 $\bmod m = 0$ 。
- 首先根据题意有 $(a_1 + a_2 + \dots + a_{k+1}) \bmod m = 0$ 且 $(a_2 + a_3 + \dots + a_{k+2}) \bmod m = 0$ 。
- 两式相减可得 $a_1 \equiv a_{k+2} \pmod m$ 。

70 分做法

- 存在 70% 的数据, $n \leq 100000$ 。
- 考虑树退化为序列的情况。
- 限制等价于任意长度为 $k+1$ 的连续区间权值和 $\bmod m = 0$ 。
- 首先根据题意有 $(a_1 + a_2 + \dots + a_{k+1}) \bmod m = 0$ 且 $(a_2 + a_3 + \dots + a_{k+2}) \bmod m = 0$ 。
- 两式相减可得 $a_1 \equiv a_{k+2} \pmod m$ 。
- 同理可得 $a_i \equiv a_{i \bmod (k+1)} \pmod m$ 。

70 分做法

- 考虑原问题，即树的情况。

70 分做法

- 考虑原问题，即树的情况。
- 类似地可以得出对于每个点 x ，设 p_x 为 x 不断往上跳 $k+1$ 个点最终到达的点，则 $a_x \equiv a_{p_x} \pmod{m}$ 。

70 分做法

- 考虑原问题，即树的情况。
- 类似地可以得出对于每个点 x ，设 p_x 为 x 不断往上跳 $k+1$ 个点最终到达的点，则 $a_x \equiv a_{p_x} \pmod{m}$ 。
- 因为是完全二叉树，故 $p_x = x$ 的点数不超过 $O(2^k)$ 。

70 分做法

- 考虑原问题，即树的情况。
- 类似地可以得出对于每个点 x ，设 p_x 为 x 不断往上跳 $k+1$ 个点最终到达的点，则 $a_x \equiv a_{p_x} \pmod{m}$ 。
- 因为是完全二叉树，故 $p_x = x$ 的点数不超过 $O(2^k)$ 。
- 对于这些点预处理出 $w_{x,y}$ 表示所有 p 为 x 的点权值调到与 y 关于 m 同余的最小代价。

70 分做法

- 考虑原问题，即树的情况。
- 类似地可以得出对于每个点 x ，设 p_x 为 x 不断往上跳 $k+1$ 个点最终到达的点，则 $a_x \equiv a_{p_x} \pmod{m}$ 。
- 因为是完全二叉树，故 $p_x = x$ 的点数不超过 $O(2^k)$ 。
- 对于这些点预处理出 $w_{x,y}$ 表示所有 p 为 x 的点权值调到与 y 关于 m 同余的最小代价。
- 那么剩下的点不再影响问题，只需要考虑这些特殊点。

70 分做法

- 首先求出每个点子树中深度的最大值，剔除因为过短不受限制的子树。

70 分做法

- 首先求出每个点子树中深度的最大值，剔除因为过短不受限制的子树。
- 那么限制转化为：每个叶子到根路径上所有点权值和 $\bmod m = 0$ 。

70 分做法

- 首先求出每个点子树中深度的最大值，剔除因为过短不受限制的子树。
- 那么限制转化为：每个叶子到根路径上所有点权值和 $\bmod m = 0$ 。
- 设 $f_{x,y}$ 表示考虑 x 的子树， x 到子树内部任意一个叶子路径上权值和 $\bmod m$ 都为 y 的最小代价。

70 分做法

- 首先求出每个点子树中深度的最大值，剔除因为过短不受限制的子树。
- 那么限制转化为：每个叶子到根路径上所有点权值和 $\bmod m = 0$ 。
- 设 $f_{x,y}$ 表示考虑 x 的子树， x 到子树内部任意一个叶子路径上权值和 $\bmod m$ 都为 y 的最小代价。
- 暴力转移即可。

70 分做法

- 首先求出每个点子树中深度的最大值，剔除因为过短不受限制的子树。
- 那么限制转化为：每个叶子到根路径上所有点权值和 $\bmod m = 0$ 。
- 设 $f_{x,y}$ 表示考虑 x 的子树， x 到子树内部任意一个叶子路径上权值和 $\bmod m$ 都为 y 的最小代价。
- 暴力转移即可。
- 时间复杂度 $O(nm + 2^k m^2)$ 。

100 分做法

- 70 分做法的瓶颈在于 $O(nm)$ 的预处理。

100 分做法

- 70 分做法的瓶颈在于 $O(nm)$ 的预处理。
- 观察 p_x 的求法，可以通过双指针优化到 $O(1)$ 。

100 分做法

- 70 分做法的瓶颈在于 $O(nm)$ 的预处理。
- 观察 p_x 的求法，可以通过双指针优化到 $O(1)$ 。
- 那么每个点对于某个 w 的贡献为一个等差数列。

100 分做法

- 70 分做法的瓶颈在于 $O(nm)$ 的预处理。
- 观察 p_x 的求法，可以通过双指针优化到 $O(1)$ 。
- 那么每个点对于某个 w 的贡献为一个等差数列。
- 而等差数列可以通过差分实现 $O(1)$ 修改， $O(m)$ 得到实际值。

100 分做法

- 70 分做法的瓶颈在于 $O(nm)$ 的预处理。
- 观察 p_x 的求法，可以通过双指针优化到 $O(1)$ 。
- 那么每个点对于某个 w 的贡献为一个等差数列。
- 而等差数列可以通过差分实现 $O(1)$ 修改， $O(m)$ 得到实际值。
- 故预处理的时间复杂度降低为 $O(n + 2^k m)$ 。

100 分做法

- 70 分做法的瓶颈在于 $O(nm)$ 的预处理。
- 观察 p_x 的求法，可以通过双指针优化到 $O(1)$ 。
- 那么每个点对于某个 w 的贡献为一个等差数列。
- 而等差数列可以通过差分实现 $O(1)$ 修改， $O(m)$ 得到实际值。
- 故预处理的时间复杂度降低为 $O(n + 2^k m)$ 。
- 总时间复杂度 $O(n + 2^k m^2)$ 。

Thank you!