

乐乐的公司

有一个有根树，每个结点可以选择 r_i 和 p_i 两种收入，和一个代价：

$$\left\lceil \frac{x_i}{1000} \right\rceil \cdot 666 \cdot i \quad (1)$$

其中 x_i 为 i 及其孩子们的收入的极差。问最大收入。 $n, r_i, p_i \leq 10^5$ 。

设 fr_u, fp_u 表示 u 子树内， u 选了 r/p 时的最大收入；转移时枚举 $\lceil \frac{x_i}{1000} \rceil$ ，然后枚举最小值，尺取法地找最大值，维护这个范围内能去的 r_v, p_v ，离散化后总复杂度 $O(100n)$ 。

树

小 D 有一棵有根树，不同节点的编号各不相同。小 Z 想知道这棵树长什么样，但是小 D 不愿意直接告诉他，小 D 只提供了 k 条以下形式的信息：给定 a, b 和 c ，小 D 的树满足编号为 a 的节点是编号为 b 的节点的 c 级祖先。小 Z 怀疑小 D 在骗他，所以他想要知道是否存在符合小 D 描述的树。

$k \leq 10^5, a, b, c \leq 10^9$ 。

法 1：从 a 向 b 连边。枚举所有没有出度的点，这些点在树上就是叶子节点；而它的闭合子图的点集，全部都位于它到根的链上，所以可以将这些点拓扑排序后，合并成一条链（去掉所有边，重新连上形如一条链的边）。依次对所有叶子结点进行这个操作并在过程中排除所有会产生非法的情况（两个点深度相同、一个点两种深度）即可判定并构造方案。

法 2：从 a 向 b 连边。依次找没有出度的点，然后将它的所有出边排序，然后从短到长连边：比如原本是 $(1 \rightarrow 2, 1)(1 \rightarrow 3, 2)(1 \rightarrow 4, 3)$ ，那么就改成： $(2 \rightarrow 3, 1)(3 \rightarrow 4, 1)$ 。像拓扑排序那样将这个方法进行下去即可。总复杂度可能是 $k^{1.5}$ 或者 $k \log k$ 的，因为边上传的次数理应不会太多，而边会被去重。

法 3：在法 2 的基础上，记下每个点上方所有祖先的深度；每次将其归并到最浅的那个祖先上，像拓扑排序那样进行下去。结合 splay 可以做到 $O(k \log k)$ 。

异或

可以看这个：<http://218.5.5.242:9019/article/828>。最关键的是：

$$\sum a_x b_x = \sum \text{FWT}(a)_x \cdot \text{iFWT}(b)_x \quad (2)$$

列出二元生成函数的式子后，以形式幂级数那一维为主元，然后运用上述式子，最后找到办法快速分治计算插值即可。

esperar

小 S 写给了小 W 共 n 个正整数 a_i ，此时，小 W 会先写出 n 个正整数 $b_i | a_i$ ，然后再写出 $d_i | b_i$ 。

如果小 W 写出的数, 满足 $\prod d_i^2 \geq \prod b_i$, 小 S 就会认为小 W 选数的方案是“资瓷”的。请问小 W 有多少种选数方案, 是被小 S 认为“资瓷”的, 结果对质数 998244353 取模。两种方案是不一样的, 当且仅当存在一个 b_i 或一个 d_j 不同。 $n \leq 100, a_i \leq 10^9$ 。

首先对于一组 d_i 满足 $\prod d_i^2 > \prod b_i$, 那么就有一组 $\prod (d_i = \frac{b_i}{d_i}) < \prod b_i$ 与其对应, 所以只要计算 $\prod d_i^2 = \prod b_i$ 的方案数即可。对于每个因子是独立的, 分别背包一下就好了。

宝石游戏

给定一棵树, 支持:

1. 单点修改。
2. 查询点 u 下方距离 l 范围内的点权的异或和。

法 1: 每 \sqrt{n} 次修改重构, 散块特别好维护 (用 dfs 序来判断一下祖先关系即可); 整块的话, 考虑把所有询问离线处理, 进子树的时候异或一次, 出子树的时候异或一次; 每次查询某个深度范围内的异或和, 采用 $O(1) - O(\sqrt{n})$ 分块维护深度为下标的异或和即可。

法 2: 采用 Linshey 分块, 对于每个块, 维护块顶到它的子树内 (在当前重链上只能经过当前块) 的每个 l 对应的异或和; 再对每个点维护其轻子树的关于 l 的异或前缀和。都使用 $O(\sqrt{n}) - O(1)$ 分块维护, 查询拆成 \sqrt{n} 个单点和整块询问即可。

法 3: 我们发现上面的东西直接上个全局平衡二叉树也是对的, 对于每个点, 维护 (在重链上只经过当前区间的点) 走向下方距离 l 范围内的前缀异或和, 查询的话拆成 \log 个区间, 总复杂度 $O(q \log^2 n)$ 。

Wechat

小 W 正在和妹子语音对话, 但是他的妹子不想理他。于是妹子每次会独立地在 $[0, 10]$ 中等概率随机实数 l , 并发送一段 l 秒的语音给他。

我们假定小 W 和他的妹子进行了 n 轮对话。我们可以用一个长度为 n 的序列 a_1, a_2, \dots, a_n 表示妹子发送的 n 段语音时长, 如果某个 $i(1 \leq i < n)$ 满足 $a_i < a_{i+1}$, 那么小 W 就会欣喜一次。

令 F_k 表示当语音段数为 n 时小 W 会欣喜 k 次的概率。求 F_0, F_1, \dots, F_{n-1} 。答案对 998244353 取模。

钦定一个集合都是小于号, 那么可以二项式反演得到 F_k 。具体地说:

$$F_k = \sum_{i \geq k}^{n-1} (-1)^{i-k} \binom{i}{k} \cdot G_{n-i} \quad (3)$$

$$G_i = [x^n](e^x - 1)^i$$

考虑二项式定理:

$$\begin{aligned} & [x^n](e^x - 1)^i \\ &= [x^n] \sum_{j=0}^i (-1)^{i-j} \cdot \binom{i}{j} \cdot e^{jx} \\ &= \sum_{j=0}^i (-1)^{i-j} \cdot \binom{i}{j} \cdot j^n \end{aligned}$$

冷静一下发现卷积一下可以得到 G_i ，然后卷一下就可以得到 F ，复杂度 $O(n \log n)$ 。