1 送你一个 DAG

定义 $f(u,k)$ 表示所有从 1 到 u 的路径长度 k 次方之和.

$$\begin{aligned} f(u,k) &= \sum_{\text{path } p \text{ from 1 to } u} \text{len}(p)^k \\ &= \sum_{\text{path } p \text{ from 1 to } u} \sum_{j=0}^{k} \left\{ {k \atop j} \right\} \text{len}(p)^{\underline{j}} \\ &= \sum_{j=0}^{k} j! \left\{ {k \atop j} \right\} \sum_{\text{path } p \text{ from 1 to } u} \binom{\text{len}(p)}{j} \end{aligned}$$

由于组合数的特殊性质 (杨辉三角), 于是可以 $O(1)$ 转移.

时间复杂度 $O(nk)$.

2 送你一棵圣诞树

这道题的最初版本是询问子树不同颜色个数, 是一道很经典的 “树比序列容易” 的题.

在树上可以这样做: 对于每一种颜色, 将这种颜色的点提取出来按 DFS 序排序; 令每个点的贡献为 1, 相邻点的 LCA 的贡献为 -1, 于是问题转化为区间求和.

修改颜色用这种做法是很好实现的, 只需对于每个颜色维护一个 `set` 即可.

询问颜色区间也很简单, 把线段树换成主席树就可以了; 有修改的话, 换成树状数组套线段树就好了.

时间复杂度 $O(n \log^2 n)$.

3 送你一颗圣诞树

直接按题意模拟一定会让 n 出现在指数中. 考虑转化问题.

考虑另一个问题: 给你一棵 n 个点的树, 现在要你将每个点编一个 $1...k$ 的编号, 满足任意两个编号相同的点之间至少存在一个编号小于它们的编号的点. 可以发现这两个问题是等价的.

转化为这个问题之后, 就至少会有一点思路了. 先随便选一个根, 设 $dp(u,S)$ 表示以 u 为根的这个子树中, 连向根的路径中 S 这个集合中的值还没有被覆盖. (也就是说对于每个 $k \in S$, 都存在一个点 v 满足 $label_v = k$ 并且 v 到 u 的路径上的每个点的 $label$ 都 $\geq k$)

但是暴力转移是 $O(nk4^k)$ 的, 无法通过此题.

仔细分析后, 我们发现如果对于点 u, 我们先枚举 $label_u = k$, 那么合并两个子树时, S_1 和 S_2 一定形如这个样子:

$$S_1 = ?????010101$$

$$S_2 = ?????000010$$

即, 第 k 位都是 0, 比 k 大的任意, 比 k 小的不能同时为 1.

那么对于比 k 小的位, 我们可以 3^k 枚举; 对于比 k 大的位, 可以用类似前缀和的方法解决.

时间复杂度 $O(nk3^k)$. 需要用一些实现上的技巧来加快程序的运行速度.