树上问题专题分享

Little09

2024年12月7日

前言

这次分享的内容总体难度是不大的，主要是帮助大家复习一遍 OI 里和树相关的一些问题的套路，trick 等，以及把一些知识点过一遍。
题目中会包含一些简单题、经典题，想必在座的同学有很多见过，不过拿出来讲一是考虑一些没见过对应套路的同学，二是见过的同学也可以加深一下影响。
希望大家都能有所收获！

LCA

默认大家已熟练掌握：倍增 LCA，树剖 LCA，欧拉序（或 DFS 序）LCA。

其实树上倍增、树链剖分、欧拉序和 DFS 序的思想很重要的，这些内容会经常出现在树上问题中，大家务必熟练掌握。

提一下 DFS 序 LCA：先求出每个点的 dfn，按 dfn 序依次写下每个点的父亲形成序列 g。若要求 u, v 的 LCA，不妨 $dfn_u < dfn_v$（相等需要特判），在 g 上从 $dfn_u + 1$ 到 dfn_v 找到 dfn 最小的数即为 LCA。使用 ST 表即可 $O(1)$ 求答案。

虚树

10 级考点。
不过大家还是要稍微掌握一点虚树的思想的，参见 NOI 2023 桂花树。

动态 DP

本质就是树剖 + 矩阵，真正的难点在代码难度。
虽然不怎么考，但是还是希望大家掌握。
相关的题目也比较无趣，不展开说了。

树的直径

性质是简单的，比如取直径两端点 u, v，则对于任意点 x，和它距离最远的点一定包含 u, v 中的一个。
还有点集直径的合并是经常用到的，可以放在线段树上维护。
边权为负数的时候请小心。

树的重心

对于树上的每一个点，计算其所有子树中最大的子树节点数，这个值最小的点就是这棵树的重心。
显然一棵树有一个或两个重心。
如果要求树的重心，可以用换根 DP 解决。

树的重心

树的重心有以下性质，具体证明不是那么重要：
1. 以树的重心为根时，所有子树的大小都不超过整棵树大小的一半。反之，如果以某个节点为根时，所有子树的大小都不超过整棵树大小的一半，那么这个节点为树的重心。

树的重心

树的重心有以下性质，具体证明不是那么重要：
1. 以树的重心为根时，所有子树的大小都不超过整棵树大小的一半。反之，如果以某个节点为根时，所有子树的大小都不超过整棵树大小的一半，那么这个节点为树的重心。
2. 树中所有点到某个点的距离和中，到重心的距离和是最小的；如果有两个重心，那么到它们的距离和一样。

树的重心

树的重心有以下性质，具体证明不是那么重要：
1. 以树的重心为根时，所有子树的大小都不超过整棵树大小的一半。反之，如果以某个节点为根时，所有子树的大小都不超过整棵树大小的一半，那么这个节点为树的重心。
2. 树中所有点到某个点的距离和中，到重心的距离和是最小的；如果有两个重心，那么到它们的距离和一样。
3. 把两棵树通过一条边相连得到一棵新的树，那么新的树的重心在连接原来两棵树的重心的路径上。

树的重心

树的重心有以下性质，具体证明不是那么重要：

1. 以树的重心为根时，所有子树的大小都不超过整棵树大小的一半。反之，如果以某个节点为根时，所有子树的大小都不超过整棵树大小的一半，那么这个节点为树的重心。
2. 树中所有点到某个点的距离和中，到重心的距离和是最小的；如果有两个重心，那么到它们的距离和一样。
3. 把两棵树通过一条边相连得到一棵新的树，那么新的树的重心在连接原来两棵树的重心的路径上。
4. 在一棵树上添加或删除一个叶子，那么它的重心最多只移动一条边的距离。

树的重心

树的重心有以下性质，具体证明不是那么重要：
1. 以树的重心为根时，所有子树的大小都不超过整棵树大小的一半。反之，如果以某个节点为根时，所有子树的大小都不超过整棵树大小的一半，那么这个节点为树的重心。
2. 树中所有点到某个点的距离和中，到重心的距离和是最小的；如果有两个重心，那么到它们的距离和一样。
3. 把两棵树通过一条边相连得到一棵新的树，那么新的树的重心在连接原来两棵树的重心的路径上。
4. 在一棵树上添加或删除一个叶子，那么它的重心最多只移动一条边的距离。
5. 一棵树的重心一定在根节点所在的重链上。一棵树的重心一定是以该树根节点重儿子为根的子树的重心的祖先。

CF685B Kay and Snowflake

求一棵树所有子树的重心。

$n \leq 3 \times 10^5$。

CF685B Kay and Snowflake

求一棵树所有子树的重心。
$n \leq 3 \times 10^5$。

一种解决方法是，利用性质 5，考虑对于点 x，考虑它重儿子的子树的重心 z 向上跳，判断重心即可。

[ZJOI2015] 幻想乡战略游戏

给定带点权、边权的树，q 次修改，每次修改一个点的点权，询问所有点到重心的带权距离和。
$n, q \leq 10^5$。（原题还有度数限制，这里不需要）

[ZJOI2015] 幻想乡战略游戏

给定带点权、边权的树，q 次修改，每次修改一个点的点权，询问所有点到重心的带权距离和。
$n, q \le 10^5$。（原题还有度数限制，这里不需要）

首先给出一个结论：本题中，重心位置和边权无关，和点权有关。
记 s_u 为 u 子树内点权和。考虑先取 u 为根，现在移动到 u 的某个儿子 v，计算相差的贡献：

$$len(u, v) \times (s_u - s_v - s_v)$$

所以如果需要 v 比 u 更优，则 $2 \times s_v > s_u$，显然最多只有一个儿子。一直走直到儿子都不优，则为答案。
显然这个点和普通不带边权树的重心位置一样。

[ZJOI2015] 幻想乡战略游戏

接下来怎么快速找到重心。相当于在树上找到一个点 x，满足 $s_{root} \leq 2 \times s_x$，且 x 最深。
这个结合普通树重心性质很容易得出。
一个简单的做法是：考虑线段树，叶子维护子树大小，其他节点维护线段树上子节点大小最大值。查询的时候在线段树上二分。
最后求答案的过程也是推推式子然后数据结构维护就行，这里不详细说了。

联考 NOI 模拟赛 T2 seafood

给定一个带点权的树，每次路径点权加，子树点权加，查询树的重心，有多个找到深度小的那个。

$n, q \leq 2 \times 10^5$。

联考 NOI 模拟赛 T2 seafood

给定一个带点权的树，每次路径点权加，子树点权加，查询树的重心，有多个找到深度小的那个。
$n, q \leq 2 \times 10^5$。

这题的修改就比幻想乡战略游戏强。
不过我们可以沿用上一题的结论：我们可以归纳得到，我们需要找到的其实就是深度最大的 u 满足 $2 \times s_u > sum$，于是我们可以思考怎么套上一题的做法，但感觉不是很好搞。每次修改如果是单点修改，那么就会有一条链上的 s 改变；而现在是子树或者路径修改，贡献不好处理。

联考 NOI 模拟赛 T2 seafood

我们尝试换个方式。

我们用线段树直接维护点权。发现如果已知重心是某个点的祖先的话，我们就可以在这条链上倍增找到重心。那么现在我们只要找到这个点就行。

使用 DFS 序将树转化为序列，那么重心的子树对应的是序列上的一个区间。注意到重心的子树的 s 一定大于一半，所以我们找到 DFS 序上第一个超过一半的位置，会发现这个点必须在重心子树内（否则重心子树永远到不了一半）。于是就解决了。

[CSP-S2019] 树的重心

给定一棵树，求出对每条边删掉这条边后分裂出的两个子树的重心编号和之和。

$n \le 3 \times 10^5$。

[CSP-S2019] 树的重心

给定一棵树，求出对每条边删掉这条边后分裂出的两个子树的重心编号和之和。
$n \le 3 \times 10^5$。

如果说上一个题是一个不好想的 trick，这个题就算是相当套路的题目，希望大家都能熟练掌握。
第一步，为了方便，我们直接取一个重心 rt 作为根。然后这个贡献形式显然只要计算每个点被作为了多少次重心即可。
考虑 x 作为重心且 $x \neq rt$，显然割掉的边不在 x 子树内。设割掉一条边后，另外一棵树的大小为 S。
设 $g_x = \max_{y \in \mathrm{son}(x)}\{s_y\}$。

[CSP-S2019] 树的重心

由于 x 为重心，所以要满足：

$$2 \times (n - S - s_x) \leq n - S$$

$$2 \times g_x \leq n - S$$

即：

$$n - 2s_x \leq S \leq n - 2g_x$$

即我们要找到对于一个 x，记 $L_x = n - 2s_x, R_x = n - 2g_x$，求有多少条可以割掉的边满足：

1. $L_x \leq S \leq R_x$。
2. 边不在 x 的子树内。

[CSP-S2019] 树的重心

这一步可以简单用数据结构维护，应该有各种办法。我的做法是先假设 S 为割掉边后的下部子树大小，那么这里用 DFS 序 + 树状数组就能搞定。然后再容斥一下错误的 S，对于 x 而言只有根到 x 的链上的 S 是错误的，我们 DFS 的时候用树状数组维护贡献就行。

以及：当 $x = rt$，我们如何统计 x 为重心的次数？

[CSP-S2019] 树的重心

设 x 的儿子中子树最大的节点为 u，次大的节点为 v。
若割掉的边在 u 的子树中，则需要满足：

$$2 \times s_v \leq n - S$$

即：

$$S \leq n - 2s_v$$

否则，需要满足：

$$2 \times s_u \leq n - S$$

即：

$$S \leq n - 2s_u$$

可以在 dfs 的时候直接维护。

[CSP-S2019] 树的重心

Tips：总之虽然题解篇幅不小，但这题本质上就是一个无脑拆贡献，然后在树上用数据结构维护一些信息的经典问题。只需要写清式子，理清思路，可以先写个暴力检查式子的正确性，然后将其改为数据结构的形式，那么这类问题就会迎刃而解。

CF1842F Tenzing and Tree

有一棵 n 个点的树，其中有 k 个点被染上了黑色。每条边都有权值，定义一条边的权值为这条边两侧黑点数量的差的绝对值。一棵树的权值是所有边的权值之和。对于 k 满足 $0 \le k \le n$，你需要确定这 k 个点，满足在染色后，这棵树的权值最大。

$n \le 5000$。

CF1842F Tenzing and Tree

有一棵 n 个点的树，其中有 k 个点被染上了黑色。每条边都有权值，定义一条边的权值为这条边两侧黑点数量的差的绝对值。一棵树的权值是所有边的权值之和。对于 k 满足 $0 \le k \le n$，你需要确定这 k 个点，满足在染色后，这棵树的权值最大。
$n \le 5000$。

Tips：毕竟是一个 CF 题，主要难度在思维难度，跟上面那个 CNOI 题还是能很明显看出差别的。

CF1842F Tenzing and Tree

有一棵 n 个点的树，其中有 k 个点被染上了黑色。每条边都有权值，定义一条边的权值为这条边两侧黑点数量的差的绝对值。一棵树的权值是所有边的权值之和。对于 k 满足 $0 \le k \le n$，你需要确定这 k 个点，满足在染色后，这棵树的权值最大。
$n \le 5000$。

Tips：毕竟是一个 CF 题，主要难度在思维难度，跟上面那个 CNOI 题还是能很明显看出差别的。
$|k - 2x|$ 绝对值你解决不掉也不好拆，那么考虑找一个黑点的重心 rt，这样一定满足 $2x \le k$，所以可以直接拆掉。x 表示子树黑点数。
于是考虑枚举这个 rt，现在形式变成了 $\sum(k - 2x)$，也就是最小化 $\sum x$，那么每个点的贡献就是到根的边数，所以贪心按深度选就行。

CF1842F Tenzing and Tree

那如果选的 rt 不是重心呢？想必是没事的，因为如果实际上是 $2x-k$ 你算成了 $k-2x$，由于后者是一个负数，所以你得到的结果只会变小，不影响最大化的结果。
“按深度选”的时候如果跑一个 BFS 就是 $O(n^2)$，如果直接把深度 sort 一下就是 $O(n^2 \log n)$，都能过。
关键 trick 是：取重心为根，然后枚举重心，但不需要确定其为重心。因为这个值在重心处最优，所以这么做就对了。

CF600E Lomsat gelral

给定一个有根树，每个点有一个颜色 c_i，对每个子树，求所有出现最多的颜色的和。

$n \leq 10^5$。

CF600E Lomsat gelral

给定一个有根树，每个点有一个颜色 c_i，对每个子树，求所有出现最多的颜色的和。
$n \leq 10^5$。

相信大家也都了解，不过还是稍微介绍一下。这个算法要求可离线且不是动态的。
同树链剖分的 dfs1，我们将其儿子分轻重。
对于每一个节点 u，我们按以下的步骤进行遍历：
1. 先遍历 u 的轻儿子，并计算答案，但不保留遍历后它对 cnt 数组的影响；
2. 遍历它的重儿子，保留它对 cnt 数组的影响；
3. 再次遍历 u 的轻儿子的子树结点，加入这些结点的贡献，以得到 u 的答案。

CF600E Lomsat gelral

实现上，我们一般会在轻儿子处理好之后清空修改。注意，第一次遍历轻儿子是递归（DFS），而第二次是直接扫子树。

可以证明时间复杂度为 $O(n \log n)$。

你不需要考虑怎么删除，因为是整体清空的。

在本题中，开桶维护子树信息即可。

CF1767F Two Subtrees

给你一棵有根树，每次给出两个子树（可能有交），求这两个子树中所有的点权的最小众数。如果一个点被两个子树覆盖算两次。

$n \leq 2 \times 10^5$。

CF1767F Two Subtrees

给你一棵有根树，每次给出两个子树（可能有交），求这两个子树中所有的点权的最小众数。如果一个点被两个子树覆盖算两次。
$n \leq 2 \times 10^5$。

这题也是被花式爆过去了，不过放在这里就给一个和 dsu on tree 相关的做法。
首先树上 dsu 有一个结论，我们先求一下 dfn，那显然每个子树是可以用 dfn 上的一段区间表示的，对 x 为根的子树记作 L_x, R_x，那么如果我们按照 dsu 的顺序依次“莫队”这些区间，也就是按照轻儿子优先访问的顺序遍历得到的出栈序，访问区间的时候如果无交就把原区间情况再挪过去，否则移动左右端点。
显然复杂度和 dsu on tree 相同，端点总移动量为 $O(n \log n)$。

CF1767F Two Subtrees

考虑结合一下这题的莫队。我们先按照上述过程 dsu on tree，然后形成一个长度为 n 的区间序列，每个询问对应两个前缀，可以挪动左右端点向相邻的区间。这其实是一个带权莫队，计算出向旁边移动的代价后按这个跑莫队，暴力移动端点即可。需要用值域分块平衡复杂度。

Tips：如果不是很理解我写的内容，课后可以参考一下对应的题解。

[USACO23JAN] Subtree Activation P

你有一棵根为 1 的树，顶点标记为 $1 \dots N$ $(2 \le N \le 2 \cdot 10^5)$ 。每个顶点最初都是关闭的。在一次操作中，你可以将一个顶点的状态从关闭状态切换到开启状态，反之亦然。输出一个满足以下两个条件的操作序列的最小可能长度。

1. 定义以顶点 r 为根的子树由所有满足 r 位于从 1 到 v 的路径上 (包括 v) ，的顶点 v 组成。每一个顶点的子树，都有一个时刻，开启状态顶点的集合恰好是该子树中的顶点。
2. 在整个操作序列之后，每个顶点都是关闭的。

[USACO23JAN] Subtree Activation P

你有一棵根为 1 的树，顶点标记为 $1 \ldots N$ $(2 \leq N \leq 2 \cdot 10^5)$ 。每个顶点最初都是关闭的。在一次操作中，你可以将一个顶点的状态从关闭状态切换到开启状态，反之亦然。输出一个满足以下两个条件的操作序列的最小可能长度。
1. 定义以顶点 r 为根的子树由所有满足 r 位于从 1 到 v 的路径上 (包括 v)，的顶点 v 组成。每一个顶点的子树，都有一个时刻，开启状态顶点的集合恰好是该子树中的顶点。
2. 在整个操作序列之后，每个顶点都是关闭的。

Tips：这个题的做法并不是 dsu on tree 相关的，但是这个题本质上其实就是让你最小化 dsu on tree 的过程中的加入次数。所以也放在这一个板块当做扩展提一下。本题也有许多不同做法，下面的做法给大家参考。

[USACO23JAN] Subtree Activation P

根据 dsu on tree，可知答案是 $O(n\log n)$ 级别的。
首先题意可以抽象问让你找一个最小的哈密顿回路（事实上可以考虑欧拉回路，因为重复经过没有影响），两点之间有一个代价。代价是两个子树点集的 XOR 大小。
考虑转化一下，建一个新的点 0，第 i 个点和 0 连边代价为 siz_i，第 i 个点个其父亲连边 $|siz_i - siz_{fa}|$。这样问题转化成：找到一个边集代价最小，满足有欧拉回路（每个点度数为 2 倍数）且点联通。
这样就可以在树上进行树形 DP 了！记 $dp_{u,0/1,0/1}$ 表示 u 这个子树内，u 的度数奇偶性，u 是否已经与 0 联通的最小代价，每次转移的时候枚举一下即可。

树哈希

判断一些树是否同构，我们需要把树映射成哈希值来存。
以下方式均用于有根树。无根树我们找到重心再做，如果有两个重心判断两个重心是否对得上。
考虑如下哈希：对于一棵以 a 为根的子树，假设儿子是 $v_1, v_2, \cdots, v_k$，定义子树的哈希 $h(a) = 1 + \sum_{1 \le i \le k} f(h(v_i))$。其中 $h(v_i)$ 是 v_i 对应子树的哈希，f 为一个随机函数。
注意到这样的写法是支持换根的，第二次 DP 时只需把子树哈希减掉即可。而且很好写。

重链剖分

最基础的内容省略了，通过重链剖分带给我们的性质有：

1. 重链内的 DFN 序是连续的。
2. 一颗子树内的 DFN 序是连续的。
3. 树上的每条路径都可以被拆分成不超过 $O(\log n)$ 条重链。

通过剖分可以将树上问题以多一个 log 的代价转化到序列上。

相信大家也做过很多题了，这里不具体讲例题了。

长链剖分

长链剖分与重链剖分有相通之处，后者是将子树大小最大的儿子作为重儿子，前者则是将子树深度最大的儿子作为重儿子。可见两者只是换了一个剖分形式。

于是可以有一些性质：

性质 1：从根节点到任意叶子节点经过的轻边条数不超过 $\sqrt{n}$，这比重链剖分 $\log n$ 稍劣一些。

性质 2：一个节点的 k 级祖先所在长链长度一定不小于 k。

性质 3：每个节点所在长链末端为其子树内最深节点。

请牢记：所有长链的长度和是 n。

最经典的应用是求 k 级祖先。具体求法相信大家是会的。

CF1009F Dominant Indices

给定一棵以 1 为根，n 个节点的树。设 $d(u,x)$ 为 u 子树中到 u 距离为 x 的节点数。对于每个点，求一个最小的 k，使得 $d(u,k)$ 最大。

$n \le 10^6$。

CF1009F Dominant Indices

给定一棵以 1 为根，n 个节点的树。设 $d(u,x)$ 为 u 子树中到 u 距离为 x 的节点数。对于每个点，求一个最小的 k，使得 $d(u,k)$ 最大。
$n \le 10^6$。

这个题是长链剖分优化 DP 的一个最简单的例子。
这类题目首先先设出二维的状态，基本形如：$dp_{u,v}$ 表示 u 子树内距离 u 为 v 的节点的某状态。本题中是 u 子树内距离 u 为 v 的节点的个数。直接合并是：

$$dp_{u,d} = \sum_{v \text{ is son of } u} dp_{v,d-1}$$

CF1009F Dominant Indices

直接合并是 $O(n^2)$。考虑长链剖分，先从重儿子继承 dp 数组，再把所有轻儿子的 dp 数组合并上去。根据性质 4，如果合并是 $O(1)$，那么这样的复杂度就是 $O(n)$ 的。

关于动态分配内存，可以用 vector 实现，也可以使用内存池分配。注意我们只在链顶申请内存，给重儿子分配的时候就用自己的指针 $+1$ 即可。

长链剖分

事实上长链剖分优化 DP 的题目不算少，联考模拟赛也出现过多次，关键思想就是在将轻儿子的 DP 状态合并上去时，我们只能访问轻链长度个数的信息，但较复杂的转移就需要一些其他操作。

以及经典问题 [十二省联考 2019] 希望，这题的模型可以套用到很多类似情况，不过这题较为复杂且代码难度也不低，大家有兴趣可以自行研究。

类似地，树上的二维 DP 优化的另一个办法是线段树合并：[NOI2020] 命运。本质上的思想也是平凡的，所以这里不多赘述。

点分治

考虑一类路径问题，点分治就是每次取树的重心 u 进行分治，将所有路径转化为经过点 u 和不经过点 u 的两类。于是分治的深度是 log 级别的，可以解决很多路径问题了。

P7782 「MCOI-Zero / AC6-M03」 Sipli Fiel

给定一个 n 个点的树，和两个常数 L, R。请对于每一个点 u 求出有多少条路径过 u 且长度 $d \in [L, R]$。

$n \leq 10^6$。

P7782 「MCOI-Zero / AC6-M03」 Sipli Fiel

给定一个 n 个点的树，和两个常数 L, R。请对于每一个点 u 求出有多少条路径过 u 且长度 $d \in [L, R]$。
$n \leq 10^6$。

Tips：标算为 1log，这里给一个常数小的 2log 点分治做法。
树上路径统计，考虑点分治。点分治的时候，一个一个子树扫过去，用树状数组维护每个子树深度为一定值的个数。
注意到本题要求路过 u 的路径个数，那么一个 trick 是，对每个点算出答案后，要累加到它的父亲节点上。正着扫完一遍还要反着扫回去。
需要注意路径有一端是根时的贡献需要特殊算。并且可以第一次扫的时候先加入根，第二次扫的时候不加入根，这样的话根的贡献只在第一次扫的时候被计算。
其实如果熟练点分治的话，看这种问题是很自然的。

P4886 快递员

n 个点的树上有 m 组点对 (a_i, b_i)，求一个点 x 满足 $dis(a_i, x) + dis(x, b_i)$ 的最大值最小。

$n, m \leq 10^5$。

P4886 快递员

n 个点的树上有 m 组点对 (a_i, b_i)，求一个点 x 满足 $dis(a_i, x) + dis(x, b_i)$ 的最大值最小。
$n, m \le 10^5$。

另一类点分治的题目。我们注意到是选取一个点 x 并最小化代价，并且对一个点 x 计算答案是 $O(n)$ 的。所以思路为先取一个 x，计算其答案，接下来我们发现可能的答案一定在 x 的某个子树内，并递归该子树，那么我们每次选择 x 为重心的话，最终只会计算 $\log n$ 次，所以复杂度为 $O(n \log n)$。
对于这道题使用上述方法，我们先取一个 x 并观察当前状态下权值最大的路径。如果它跨过 x 那目前就是最小值，如果有两条最大路径满足分居 x 的不同子树，那可以发现调整也不会更优。否则就在某个 x 的子树，递归这个子树即可。

点分树

点分树是通过更改原树形态使树的层数变为稳定 $\log n$ 的一种重构树。将每次找到的重心与上一层的重心缔结父子关系即可。

一般对于每一道点分树上的计数问题，都需要注意如何去掉每个节点与其父节点之间重叠的贡献。我们可以维护两个数据结构，例如上面的题，一个维护 x 子树内距离 x 的距离 $\leq d$ 的点权和，另一个维护 x 子树内距离 fa_x 的距离 $\leq d$ 的点权和。每次查询的时候用这两个数据结构减一下就行。

对于最优化问题，通常不用考虑上述情况，因为一般来自相同子树的情况是不优的。不过还是请具体问题具体分析。

联考 NOI 模拟赛 T2 偷笑

现在 Y 有一棵 n 个节点的树，树上的每条边都有长度，每个节点上有一盏灯，初始时每个节点上的灯都是关闭的。

接下来有 Q 个时刻，每一时刻会有如下三种操作之一发生：

1. **query** 操作。给定一个节点 q_i，询问到离它最近的一盏打开的灯的距离。
2. **change** 操作。给定一个节点 u_i，反转 u_i 上的灯的状态。
3. **modify** 操作。给定三个整数 a_i, b_i, w_i，将连接 a_i 和 b_i 的边的长度改为 w_i。

Y 想要你给出所有 **query** 操作的答案。

$n, Q \leq 10^5$。

联考 NOI 模拟赛 T2 偷笑

注意到这道题不需要减掉计算父亲的时候来自原先子树的贡献，因为这里是最优化，而来自原先子树一定不优。考虑对每个询问在点分树上暴力跳祖先，那么我们对于每个点 u，需要实时维护子树内每个点到它的距离和它到最近的亮着的灯的最小距离。

我们会发现是可以维护的，每个点建一个线段树，其中下标按这个连通块遍历的 DFS 序建。前两个操作暴力跳祖先维护，第三个操作在每个涉及到的节点上是一个区间加。

集训队互测 2023 不跳棋

给一棵有 n 个节点的树，初始每个节点上都有一个棋子。共有 $n-2$ 次操作，每次拿走树上的一个棋子。
请在每次操作后输出最小的任意两个不同棋子的树上距离以及最小距离的点对数量，强制在线。
$n \le 5 \times 10^5$。

集训队互测 2023 不跳棋

给一棵有 n 个节点的树，初始每个节点上都有一个棋子。共有 $n-2$ 次操作，每次拿走树上的一个棋子。
请在每次操作后输出最小的任意两个不同棋子的树上距离以及最小距离的点对数量，强制在线。
$n \leq 5 \times 10^5$。

和上一题类似地，我们考虑点分树。在每个分治重心处统计经过该重心的路径条数。注意到仍然是最优化，所以对于两个来自同一个子树的路径是一定更劣的，所以把它们统计进来是可以的。
那么我们开一个大小为连通块深度的桶，维护最小指针和次小指针，每次一定是最小和次小的位置得到该连通块的信息。
修改的时候在点分树上跳祖先修改即可。复杂度 $O(n \log n)$。

树形背包

直接暴力背包相信大家是会的，就是每次合并 u, v 时 $O(siz_u \times siz_v)$ 合并，那么总复杂度为 $O(V^2)$。
下面主要说一下 DFS 背包。

LOJ160 树形背包

有 N 个物品，编号分别为 $1 \dots N$。物品 i 的重量为 w_i，价值为 v_i。
给出每个物品依赖于哪个物品。我们用 $d_i = j(i > j > 0)$ 表示：如果要选取物品 i，就必须先选取物品 j。另外，我们用 $d_i = 0(i > 0)$ 表示：该物品不依赖于任何物品。保证每个物品最多只依赖一个物品。保证依赖关系合理，不会出现环。背包最多能装载的重量为 W，请问背包中最多能装入多大价值的物品。
背包最多能装载的重量为 W，请问背包中最多能装入多大价值的物品。
$N \times W \leq 6 \times 10^7$。

LOJ160 树形背包

直接做背包：定义 $dp_{i,j}$ 表示第 i 个节点重量为 j 的最大价值。每次转移是一次背包合并，$O(W^2)$。总复杂度 $O(nW^2)$。
考虑一种按照 DFS 序做背包的方式。按照 DFS 倒序考虑每个节点，定义 $dp_{i,j}$ 表示已经考虑了第 i 个及 DFS 在它后面的所有节点，在这些节点中选出的重量为 j 的最大价值。考虑每个点 i 的转移：如果不选，那么也不能选 i 的子树，所以由 $i + siz_i$ 转移；否则选的话，由 $i + 1$ 转移。

来源丢失的题目

有一棵 n 个点的树，每个点 i 有一个物品，重量为 $V_i(V_i <= m)$。对 $k = 1 \sim m$ 求出，树上有多少个连通块物品重量和为 k，模 $10^9 + 7$。

$n \leq 200, m \leq 50000$。

来源丢失的题目

暴力做是对每个点 i 求 $dp_{i,j}$ 表示以 i 为根的连通块重量和为 j 的方案数。那么每次又要做背包合并，复杂度为 $O(nm^2)$。
如果必须包含 1，那就是树上 DFS 序背包。枚举连通块的根，做树上依赖背包，复杂度为 $O(n^2m)$。
考虑 dsu。每次在做节点 u 的时候，先直接继承重儿子的 dp 数组，再遍历所有轻儿子子树。根据 dsu on tree，这样的遍历量是 $O(n\log n)$ 的。每次遍历的时候，就按照上述做树上依赖背包的方法做，就可以对每个轻儿子 v，算出包含 v 的连通块的 dp 数组。这里只要做 $O(n\log n)$ 次将一个物品加入背包就行，复杂度为 $O(nm\log n)$。
更简单的想法是点分治，然后做 DFS 序背包。不过 dsu 可以求出每个点为根的答案。

牛客186E 旅行

给定一棵 N 个点的有点权的树，另外给定一个 K。M 次询问树上一条链中，有多少种方案选一个点集使得点权和为 K 的倍数。

$N, M \leq 2 \times 10^5$，$K \leq 50$。

牛客186E 旅行

给定一棵 N 个点的有点权的树，另外给定一个 K。M 次询问树上一条链中，有多少种方案选一个点集使得点权和为 K 的倍数。

$N, M \le 2 \times 10^5$，$K \le 50$。

树上的链可以先考虑树剖，然后大概要背包合并，又要在线段树上搞，复杂度大概是 $O(nK^2 \log^2 n)$，非常爆炸。
考虑点分治。首先在点分树上找到一条路径的 LCA，我们在这个点上计算这条路径的答案。考虑在算点 u 的时候，遍历一遍子树，对子树里的每个点 v，算出 $v \to u$ 路径的背包。这里不存在背包合并，所以这一步是 $O(nK \log n)$ 的。算答案的时候似乎要背包合并，但其实由于只关心最后 $\bmod K = 0$ 的部分，所以直接枚举一下就行了，不用合并。这样复杂度是
$O(nK \log n + qK + q \log n)$ 的。

一些其他树上 DP

事实上列出的这些题跟背包还是脱不开关系，不过会更加综合一些。

[ICPC2022 Jinan R] DFS Order 2

给定一棵 n 个点的树，对每个点 i 求在所有 DFS 序其是第 j 个访问的点的方案数。

$n \leq 500$。

[ICPC2022 Jinan R] DFS Order 2

给定一棵 n 个点的树，对每个点 i 求在所有 DFS 序其是第 j 个访问的点的方案数。
$n \le 500$。

先对每个子树求出其 DFS 序个数。
考虑 DP，记 $dp_{i,j}$ 表示做到第 i 个点，他在 DFS 序的第 j 位的方案数。那么对于 $dp_u \to dp_v$ 的转移，考虑 u 除了 v 以外的其他儿子的子树，要么在 v 之前被访问要么在 v 之后，我们记录有多少个儿子子树是在 v 之前的，以及这些子树的大小和。
假设有 x 个 v 之前的和 y 个 v 之后的，那么系数需要乘上 $x!y!$。
我们对每个 u 都先枚举 v，然后对除 v 外的其他子树做上述的二维背包。总复杂度为 $O(n^4)$。

[ICPC2022 Jinan R] DFS Order 2

显然上述做法每次重新跑背包太劣了，但是如果对前后缀处理背包，合并复杂度又爆炸了。
所以我们可以使用撤销背包。先把所有儿子的子树都加入，每次删除的时候直接怎么加的就怎么删就行。
那么复杂度为 $O(n^3)$。

P8935 [JRKSJ R7] 茎

你有一棵 n 个点的根节点为 1 的有根树，现在你要对这棵树进行剪枝，每次你可以选择一个还未被剪掉的节点 u 进行操作，然后剪掉 u 的子树所有点（包括 u）。当且仅当你剪掉 1 时，操作停止。

你需要在第 k 次剪枝时对 x 进行操作。求有多少种不同的操作序列。

$1 \le k, x \le n \le 5000$。（原题为 500）

P8935 [JRKSJ R7] 茎

题意的“茎”可以形象理解为 1 到 x 的路径。
首先可以套路做一个树形背包，定义 $dp_{u,v}$ 表示 u 子树内剪掉了 v 个节点的方案数，合并的时候乘一个组合数就行了。
考虑从根开始沿着茎做 DP。但是状态不好搞。
考虑一个构造最终序列的方式：从 1 开始一路走到 x，维护一个序列，每次处在 u 时，先把 u 子树不在路径上的儿子的序列插入这个序列，然后再考虑 u，u 如果不插就直接往下，如果插的话就是在序列里选一段后缀删掉（因为之后做的这个子树每个点必须在它前面）。注意 1 和 x 都必须插。
对这个东西做 DP 即可，设 $g_{u,v}$ 表示做到 u 当前序列长度 v，每次先把其他子树用 dp 给合并了，再沿着茎往下做。答案就是 $g_{x,k-1}$。

[ZJOI2022] 树

求有多少种两棵树满足，第一棵树的 $fa_i < i$，第二棵树的 $fa_i > i$，且每个点恰好在其中一棵树内是叶子。答案对 M 取模。

$n \le 500$

[ZJOI2022] 树

求有多少种两棵树满足，第一棵树的 $fa_i < i$，第二棵树的 $fa_i > i$，且每个点恰好在其中一棵树内是叶子。答案对 M 取模。

$n \le 500$

设 $f(S)$ 表示第一棵树的非叶子集合恰好为 S 时构造第一棵树的方案数，设 $g(T)$ 表示第二棵树的非叶子集合恰好为 T 时构造第二棵树的方案数。
那么答案为

$$Ans = \sum_{S\cap T=\varnothing, S\cup T=\{1,2,\cdots,n\}} f(S)g(T)$$

直接求这个不太好求，我们考虑容斥。

[ZJOI2022] 树

设 $f'(S)$ 表示第一棵树的非叶子集合包含于 S 时构造第一棵树的方案数，设 $g'(T)$ 表示第二棵树的非叶子集合包含于 T 时构造第二棵树的方案数。

$$\begin{aligned} Ans &= \sum_{S\cap T=\varnothing, S\cup T=\{1,2,\cdots,n\}} f(S)g(T) \\ &= \sum_{S\cap T=\varnothing, S\cup T=\{1,2,\cdots,n\}} \sum_{S'\subseteq S, T'\subseteq T} f'(S')g'(T')(-1)^{|S|-|S'|+|T|-|T'|} \\ &= \sum_{S'\cap T'=\varnothing} f'(S')g'(T')(-1)^{n-|S'|-|T'|}2^{n-|S'|-|T'|} \\ &= \sum_{S'\cap T'=\varnothing} f'(S')g'(T')(-2)^{n-|S'|-|T'|} \end{aligned}$$

其中倒数第二个等号的理由是：不属于 S' 和 T' 的那些元素可能在 S 中，也可能在 T 中。

[ZJOI2022] 树

然后可以 DP，我们设 $dp_{i,j,k}$ 表示确定了 $[1,i]$ 中的点在 S' 和 T' 中的情况，$|\{1,\cdots,i\}\cap S'|=j$，$|\{i+1,\cdots,n\}\cap T'|=k$，且我们已经为 $(1,i]$ 选好了第一棵树中的父亲，为 $[1,i)$ 选好了第二棵树中的父亲时，这些父亲的方案数。

转移时，分类讨论 i 在 S' 和 T' 中的情况，然后确定 i 在第一棵树中的父亲和 $i-1$ 在第二棵树中的父亲：

1. i 属于 S'，$dp_{i-1,j,k}$ 转移到 $dp_{i,j+1,k}$，系数为 $j\times k$。
2. i 属于 T'，$dp_{i-1,j,k}$ 转移到 $dp_{i,j,k-1}$，系数为 $j\times k$。
3. i 两个都不属于，$dp_{i-1,j,k}$ 转移到 $dp_{i,j,k}$，系数为 $-2\times j\times k$。

CF917D Stranger Trees

给定一棵有 n 个节点的无权无向树。
求对于这 n 个点，以及每个 $k = 0, 1, 2 \ldots n-1$，有多少棵由这 n 个点之间的边构造成的树，与给定的树恰好有 k 条边重复。
$n \leq 5000$（原题 $n \leq 100$）。

CF917D Stranger Trees

给定一棵有 n 个节点的无权无向树。
求对于这 n 个点，以及每个 $k=0,1,2...n-1$，有多少棵由这 n 个点之间的边构造成的树，与给定的树恰好有 k 条边重复。
$n \leq 5000$（原题 $n \leq 100$）。

二项式反演，恰好转钦定。
那么假设我们钦定了 k 条边，剩下的点构成 $n-k$ 个连通块。
接下来就是随便连把森林连成树，这是经典结论了：连通块 siz 乘积乘以 n^{k-2}（这个结论来自 Prufer 序列，可以参考 OI Wiki）。

CF917D Stranger Trees

当然我们不能暴力钦定，我们需要在树上做 DP 的时候钦定。那么显然就可以得到一个 $O(n^3)$ 的 DP，考虑记录当前连通块有多少点，目前钦定了多少条边，每次选择钦定或不钦定的时候算一下贡献。
不过这个树上断边，贡献为连通块的乘积也是经典问题，考虑组合意义就是每个连通块选一个关键点。那么我们再记录一维表示当前连通块有没有选择关键点，就可以 $O(n^2)$ 做了。最后转成恰好的反演也是 $O(n^2)$ 的。

[ARC087F] Squirrel Migration

给定一个 n 个节点的树，求一个 $1 \sim n$ 的排列 $p_1, ...p_n$，使得 $\sum dis(i, p_i)$ 最大。求满足该条件的排列的个数。
$n \leq 5000$。

[ARC087F] Squirrel Migration

给定一个 n 个节点的树，求一个 $1 \sim n$ 的排列 $p_1, ...p_n$，使得 $\sum dis(i, p_i)$ 最大。求满足该条件的排列的个数。
$n \leq 5000$。

考虑找一个重心，那么如果每个点都对应的是重心的不同子树的话，容易发现这样是答案的上界，证明可以考虑每条边被经过的次数最多是两边的子树大小的较小值。
不过重心这个点归任意子树都行。

[ARC087F] Squirrel Migration

那么就变成了每个点有一个颜色，现在要在二分图上匹配，每个点不能匹配和自己颜色相同的点（不过重心也可以匹配自己）。这就转化成了一个经典容斥问题，考虑我们要求的是恰好 0 个人匹配了自己颜色的，那么我们转换成钦定，然后依次考虑每个颜色做背包就行。

复杂度 $O(n^2)$。

CF724F Uniformly Branched Trees

给定三个数 n, d, mod，求有多少种 n 个点的不同构的树满足：除了度数为 1 的结点外，其余结点的度数均为 d。答案对质数 mod 取模。

$1 \le n \le 1000$，$2 \le d \le 10$，$10^8 \le mod \le 10^9$。

CF724F Uniformly Branched Trees

给定三个数 n, d, mod，求有多少种 n 个点的不同构的树满足：除了度数为 1 的结点外，其余结点的度数均为 d。答案对质数 mod 取模。
$1 \le n \le 1000$，$2 \le d \le 10$，$10^8 \le mod \le 10^9$。

一种多维 DP 的无标号树计数的思维方式。
首先这是无根树，那我们总得定根，我们直接取重心为根，先考虑重心唯一的情况。

CF724F Uniformly Branched Trees

于是我们根据前面可以得到一个 DP 状态：$dp_{i,j,k}$ 表示 i 个点，根节点度数为 j，每个子树大小不超过 k 的树的数量。
我们枚举有几个大小为 k 的子树就行。则我们枚举有 t 棵这样的子树，那么除去这 t 棵子树，剩下部分的状态即为 $f_{i-t\times k,j-t,k-1}$。而由于子树之间是可以相同的。所以这 t 棵子树的方案数，实际就是从 $f_{k,d-1,k-1}$ 这么多数量中选出 t 棵且可以重复的方案数，因此可以直接用组合数表达出来。
然后一个重心就做完了，对于两个重心的情况，我们发现如果有一条边连接了两个不同的大小为 $\frac{n}{2}$ 的子树，那么这种情况会多算一次，所以我们直接减掉就行。

bzoj.4771 七彩树

给定一棵 n 个节点的树，每个节点有一个颜色 c_i，m 个询问，每次询问 x 子树内，与 x 距离不超过 d 的所有节点的不同颜色数。强制在线。

$n, m \leq 10^5$。

bzoj.4771 七彩树

前置 trick：从 x 往上跳祖先，直到一个点的子树内存在至少一个目标点。不需要直接二分，可以考虑 x dfs 序前后两个，跟 x 取一下 LCA，取深度大的那个。同样的套路也用于 CF1648E Air Reform。

虽然强制在线，不过我们可以先想离线。考虑可以依次加入每个深度的点，然后考虑算贡献。想必每个点 x 应该贡献给的是它向上一段链，假设这个点颜色是 c_x，就沿着祖先的链跳直到 c_x 出现在子树内，记此时点是 v，然后给 x $+1$ 给 v -1。这样就贡献到了不重复的颜色。如果强制在线，就用可持久化线段树维护这个，每次在对应的深度上查询一下 dfs 序一段区间的 sum。

[AGC023F] 01 on Tree

给出一棵 n 个节点的树，以及一个空序列。每个节点上有一个取值在 $\{0,1\}$ 中的数。每次你可以选择没有父亲节点的点删除，并且将这个节点上的数字放在当前数列末尾。请你求出这个数列可能得到的最小逆序对数。

$n \leq 2 \times 10^5$。

[AGC023F] 01 on Tree

类似的题：[HNOI/AHOI2018] 排列。还有很多不在这里列出了。

这种题第一次做觉得很厉害，第二次就成套路了。
给你一个树让你按拓扑序选点，最优化某个代价。我们把这个方法叫做 Exchange Argument。
考虑在最终序列中整体交换两个部分 A, B 的贡献，计算出对于两个部分，谁在前面更优，并给每个部分一个权值表示按权值排序会尽可能的优。在树上维护这个过程，每次取出剩余的权值最优的块，将其与父亲合并，并重新计算权值。
在本题中经过简单计算可以发现，权值即为 0 的个数在总个数中的占比，因此模拟这个过程就行。

联考 NOI 模拟赛 T1 颜色

一个 n 个点的树，树的每条边初始都有一种颜色。
对于树上的简单路径 (x, y)，认为其是完美的，当且仅当这条路径上不存在两条颜色相同的边。
接下来进行 q 次操作：
1 x y 将第 x 条边颜色改为 y。
2 x y 查询 x 到 y 路径是不是完美的。如果是完美的输出 'YES'，否则输出 'NO'。
$n, q \leq 10^5$。

联考 NOI 模拟赛 T1 颜色

不同于数颜色的是，这里只需要判断颜色是否全部出现，那我们需要利用一下这里的性质，可以转化为：求出现次数为奇数次的颜色个数。

假设答案是 c，注意到当全部出现的时候，得到的结果是 c，否则一定小于 c，但具体是多少我们不用管。

联考 NOI 模拟赛 T1 颜色

然后再利用奇数次的性质，可以进行差分之类的东西。考虑欧拉序，注意到我们可以在欧拉序上用一段区间中仅出现一次的表示树上的一条链，而两次的本来就会被 XOR 掉，所以跑一个欧拉序之后就变成了序列问题，单点修改，区间查询出现次数为奇数次的颜色个数，强制在线。这个问题直接分块，每个块维护前缀颜色 bitset 即可，复杂度是 $O(n\sqrt{n}+\frac{nq}{w})$。

联考 NOI 模拟赛 T1 颜色

事实上也可以树分块（随机撒点）做。这样不用欧拉序。我们维护每个关键点到根的 bitset，那么查询只要跳到关键点就行。修改也是容易的。如果比较熟悉树分块，那这种做法可能更好想一点。

联考 NOIP 模拟赛 T2 失眠

给定一个 n 个点的树，第 i 个点有权值 a_i，对于 $0 \le x \le \sum a_i$，求权值和为 x 的独立集个数，对 10^9+7 取模。
$n \le 100, \sum a_i \le 10^5$。

联考 NOIP 模拟赛 T2 失眠

给定一个 n 个点的树，第 i 个点有权值 a_i，对于 $0 \le x \le \sum a_i$，求权值和为 x 的独立集个数，对 10^9+7 取模。
$n \le 100, \sum a_i \le 10^5$。

由于值域很大，如果我们要进行背包合并就必须采用卷积，但是我们不能用卷积，所以我们想找一个不需要背包合并的做法。

联考 NOIP 模拟赛 T2 失眠

考虑一种特殊情况，即如果树是一棵完全二叉树时怎么做。
由于不能用背包合并，所以我们必须用一种特殊的顺序进行背包，同时能够记录独立集的信息。不难发现对于完全二叉树而言，dfs 序就是一个合理的顺序。我们按照 dfs 序进行背包，同时记录当前点的所有祖先是否被选择，这样在 dp 完这个子树前往下一个子树时，我们只需要用到其所有祖先是否被选择这个信息。背包的同时，对于每个状态，状压记录所有祖先的选择情况，由于树的深度为 $\log n$，因此复杂度为 $O(n^2m)$。

联考 NOIP 模拟赛 T2 失眠

对于一般的情况，我们也希望记录较少量的信息，也就是说我们希望“将树高控制在 $\log n$” 的级别。不难想到的做法是建出点分树，按照点分树的 dfs 序进行背包。
在一个点分树的子树内进行 dp 之后还有用的信息只会是在点分树上的祖先是否被选择（一个点在原树上的相邻点要么在点分树的子树内，要么是点分树上的祖先，对于子树内的点考虑在那些点进行计算，而对于祖先点我们进行了状压）。点分树的树高也是 $\log n$，因此复杂度也能做到 $O(n^2m)$。已经足够通过本题了。

联考 NOIP 模拟赛 T2 失眠

事实上还可以更好。
我们想到重链剖分，如果我们按照先遍历轻儿子，后遍历重儿子的顺序进行背包，那么可能与接下来遍历到的点直接有连边的点只有每一条重链的链底。因此对原树进行重链剖分，在背包过程中记录当前点往上的所有重链链底的选择情况，按照先确定自己，再遍历轻儿子，最后遍历重儿子的顺序进行背包，这样的复杂度是 $O(\sum 2^{dep_u} \cdot m)$ 的，其中 dep_u 表示 u 上面的重链个数。

联考 NOIP 模拟赛 T2 失眠

感性理解这个东西在完全二叉树的时候跑的最慢。分析一下复杂度，令 $f(x)$ 表示完全二叉树深度是 x 的时候这个东西的值，有 $f(x) = 3f(x-1) + 1 = O(3^x)$，就是我们钦定左儿子是重儿子，右儿子是轻儿子，那么 dep_u 就是往上跳的时候跳右儿子的次数，那么右儿子的值集体 $\times 2$。因此复杂度被优化至 $O(n^{\log_2 3} m)$。

联考 NOIP 模拟赛 T2 失眠

关于本题的一些其他想法：事实上可以把本题的背包稍作修改：每个树上的点有 a_i 个物品，每个物品重量 b_i 价值 c_i，要求选择至少一个物品的位置构成独立集，做一个最优化的背包，这样原本的卷积变成了一般形式的 min + 卷积，就不能 NTT 了，而本题的做法（点分树和树剖）本质上是一样的，只需要把背包的过程稍作修改即可。

结语

谢谢大家!