树上问题

李佳衡

实验舱科学辅导中心

2020 年 8 月 13 日

树

- n 个点、$n-1$ 条边的无向连通图；
- 无向无环连通图；
- 任意两点间有且仅有一条简单路径的无向图；
- 任何边均为桥的连通图；
- 没有圈，且在任意不同两点间添加一条边后所得图含唯一一个圈的图。

树上 DFS

从根（或无根树的任意一点）开始，依次递归搜索每个儿子，同时维护当前点的信息。

树上 DFS

从根（或无根树的任意一点）开始，依次递归搜索每个儿子，同时维护当前点的信息。

DFS 序

树上 DFS 时初次访问点的序列。

树上 DFS

从根（或无根树的任意一点）开始，依次递归搜索每个儿子，同时维护当前点的信息。

DFS 序

树上 DFS 时初次访问点的序列。

- 树上每个点的子树在 DFS 序上都是一个连续的区间。

树上 DFS

从根（或无根树的任意一点）开始，依次递归搜索每个儿子，同时维护当前点的信息。

DFS 序

树上 DFS 时初次访问点的序列。

- 树上每个点的子树在 DFS 序上都是一个连续的区间。

欧拉序

树上 DFS 时初次访问及儿子回溯点的序列。

树上 DFS

从根（或无根树的任意一点）开始，依次递归搜索每个儿子，同时维护当前点的信息。

DFS 序

树上 DFS 时初次访问点的序列。

- 树上每个点的子树在 DFS 序上都是一个连续的区间。

欧拉序

树上 DFS 时初次访问及儿子回溯点的序列。

入栈出栈序（括号序）

树上 DFS 时点入栈、出栈组成的序列。

[BZOJ 3653] 谈笑风生

给定一棵 n 个点的有根树，另有 q 次询问，每次询问给定 a、k，求有多少个点对 (b, c) 满足 a、b、c 两两不同，a、b 都是 c 的祖先且 a、b 间距离不超过 k。

$n, q \leq 3 \times 10^5$。

[BZOJ 3653] 谈笑风生

给定一棵 n 个点的有根树，另有 q 次询问，每次询问给定 a、k，求有多少个点对 (b, c) 满足 a、b、c 两两不同，a、b 都是 c 的祖先且 a、b 间距离不超过 k。

$n, q \leq 3 \times 10^5$。

解

有两种情况。当 b 是 a 的祖先时，b 在 a 的 1 至 k 级祖先中任选，c 在 a 的子树内任选。

当 b 在 a 子树内时，要求 $\mathrm{dep}_b \leq \mathrm{dep}_a + k$，$c$ 在 b 的子树内任选。可以用主席树（或线段树合并）维护 a 子树内、dep 在某个区间限制内的 size 和。

时间复杂度 $O((n+q)\log n)$。

[BZOJ 4034] 树上操作

给定一棵 n 个点的有根树，有点权，维护 m 次操作：

1. 给某个点 x 的点权增加 a;
2. 给某个点 x 子树内所有点的点权增加 a;
3. 询问某个点 x 到根路径的点权和。

$n, m \leq 10^5$，输入权值不超过 10^6。

[BZOJ 4034] 树上操作

给定一棵 n 个点的有根树，有点权，维护 m 次操作：

1. 给某个点 x 的点权增加 a;
2. 给某个点 x 子树内所有点的点权增加 a;
3. 询问某个点 x 到根路径的点权和。

$n, m \leq 10^5$，输入权值不超过 10^6。

解

求出这棵树的入栈出栈序，则操作 1、2 分别对应单点、区间修改；而操作 3 实际上为从根出发到 x 的入栈出栈权值之差，都可以用线段树维护。

时间复杂度 $O(m \log n)$。

[BZOJ 1095] Hide 捉迷藏

给一棵 n 个点的树，点有黑色、白色两种。维护 q 次操作：

- 改变点 i 的颜色；
- 求所有黑色点对距离的最大值。

$n \leq 10^5$，$m \leq 5 \times 10^5$。

[BZOJ 1095] Hide 捉迷藏

给一棵 n 个点的树，点有黑色、白色两种。维护 q 次操作：

- 改变点 i 的颜色；
- 求所有黑色点对距离的最大值。

$n \leq 10^5$，$m \leq 5 \times 10^5$。

解

考虑树的括号序，两点的距离即为括号序区间内未匹配的括号数。

[BZOJ 1095] Hide 捉迷藏

给一棵 n 个点的树，点有黑色、白色两种。维护 q 次操作：

- 改变点 i 的颜色；
- 求所有黑色点对距离的最大值。

$n \le 10^5$，$m \le 5 \times 10^5$。

解

考虑树的括号序，两点的距离即为括号序区间内未匹配的括号数。

考虑用线段树维护，合并一定形如")))(((" + "))((" 合并成")))(((" 设四部分括号分别有 a、b、c、d，则合并后两部分括号分别有 $a + \max\{c - b, 0\}$、$\max\{b - c, 0\} + d$ 个。

[BZOJ 1095] Hide 捉迷藏

给一棵 n 个点的树，点有黑色、白色两种。维护 q 次操作：

- 改变点 i 的颜色；
- 求所有黑色点对距离的最大值。

$n \leq 10^5$，$m \leq 5 \times 10^5$。

解

考虑树的括号序，两点的距离即为括号序区间内未匹配的括号数。

考虑用线段树维护，合并一定形如"))((((" + "))((" 合并成")))(((" 设四部分括号分别有 a、b、c、d，则合并后两部分括号分别有 $a + \max\{c - b, 0\}$、$\max\{b - c, 0\} + d$ 个。

距离即为 $a + |b - c| + d$，讨论绝对值，分别维护后缀 $a + b$、$a - b$，前缀 $d + b$、$d - b$ 的最大值即可。

时间复杂度 $O(m \log n)$。

最近公共祖先（Lowest Common Ancestor，LCA）

树上两点的所有公共祖先中深度最大的点。

最近公共祖先（Lowest Common Ancestor，LCA）

树上两点的所有公共祖先中深度最大的点。

朴素算法

每次从两点中找到深度较大的点并向上跳至父亲，直至两点相遇。

最坏时间复杂度 $O(n)$。

最近公共祖先（Lowest Common Ancestor，LCA）

树上两点的所有公共祖先中深度最大的点。

朴素算法

每次从两点中找到深度较大的点并向上跳至父亲，直至两点相遇。

最坏时间复杂度 $O(n)$。

倍增算法

预处理出每个点的 2^k 级祖先，利用任何一个数都可以二进制表示的性质，先将较深的点跳至与令一点同一层。然后从高到低试验，如果两点祖先不相遇则一起向上跳，最后两点的公共父亲即为答案。

预处理时间复杂度 $O(n \log n)$，单次时间复杂度 $O(\log n)$。

最近公共祖先

欧拉序 RMQ 算法

求出树的欧拉序，发现两点间经过了 LCA，并且没有经过比 LCA 深度更小的点，因此只需求出欧拉序区间内深度最小的点即可，可以使用 RMQ 算法解决。

预处理时间复杂度 $O(n \log n)$ 或 $O(n)$，单次时间复杂度 $O(1)$。

最近公共祖先

欧拉序 RMQ 算法

求出树的欧拉序，发现两点间经过了 LCA，并且没有经过比 LCA 深度更小的点，因此只需求出欧拉序区间内深度最小的点即可，可以使用 RMQ 算法解决。

预处理时间复杂度 $O(n \log n)$ 或 $O(n)$，单次时间复杂度 $O(1)$。

重链剖分算法

预处理时间复杂度 $O(n)$，单次最坏时间复杂度 $O(\log n)$，常数因子优秀。

[BZOJ 1787] Meet 紧急集合

给定一棵 n 个点的树，有 m 次询问，每次给定三个点 x、y、z，要找到一个集合点，使得 x、y、z 到集合点距离和的最小。

$n, m \leq 5 \times 10^5$。

[BZOJ 1787] Meet 紧急集合

给定一棵 n 个点的树，有 m 次询问，每次给定三个点 x、y、z，要找到一个集合点，使得 x、y、z 到集合点距离和的最小。

$n, m \le 5 \times 10^5$。

解

求出两两间 LCA，三者中必有两者相同，另一个即为所求的集合点。

时间复杂度 $O((n+m)\log n)$。

[BZOJ 3910] 火车

给定一棵 n 个点的树，你需要从 a 点开始，依次走最短路径经过给定的 m 个点，求有多少边被走过。

$n \leq 5 \times 10^5$，$m \leq 4 \times 10^5$。

[BZOJ 3910] 火车

给定一棵 n 个点的树，你需要从 a 点开始，依次走最短路径经过给定的 m 个点，求有多少边被走过。

$n \leq 5 \times 10^5$，$m \leq 4 \times 10^5$。

解

用并查集维护每个点第一个未访问过的祖先位置，每次移动时两点同时向上跳至 LCA 并将并查集合并，每个点只会被额外访问一次。

时间复杂度 $O(n \log n)$。

树上差分

序列差分

对一个长为 n 的序列进行 m 次操作，每次操作对序列的一个区间全体加上一个数，求最终的序列。

树上差分

序列差分

对一个长为 n 的序列进行 m 次操作，每次操作对序列的一个区间全体加上一个数，求最终的序列。

对序列差分，则每次在 $[l, r)$ 处 $+a$ 相当于差分在 l 处 $+a$、在 r 处 $-a$，最后求前缀和即可。

时间复杂度 $O(n+m)$。

树上差分

在一棵 n 个点的树上进行 m 次操作，每次操作对一个点到根全体加上一个数，求最终的序列。

对树上差分（每个点减去其父亲），则对于一个点到根 $+a$ 相当于差分在该点处 $+a$，最后子树求和即可。

时间复杂度 $O(n+m)$。

树上差分

序列差分

对一个长为 n 的序列进行 m 次操作，每次操作对序列的一个区间全体加上一个数，求最终的序列。

对序列差分，则每次在 $[l, r)$ 处 $+a$ 相当于差分在 l 处 $+a$、在 r 处 $-a$，最后求前缀和即可。

时间复杂度 $O(n+m)$。

树上差分

在一棵 n 个点的树上进行 m 次操作，每次操作对一个点到根全体加上一个数，求最终的序列。

对树上差分（每个点减去其父亲），则对于一个点到根 $+a$ 相当于差分在该点处 $+a$，最后子树求和即可。

时间复杂度 $O(n+m)$。

- 也可以解决树上路径的操作，将路径按照 LCA 拆开即可。

[UOJ 150] 运输计划

给定一棵 n 个点、有边权的树，再给定 m 条路径，路径花费时间为其边权和。你可以将某一条边权设为 0，使得这 m 条路径同时出发，全部到达的时刻最早。

$n, m \leq 3 \times 10^5$，边权不超过 1000。

[UOJ 150] 运输计划

给定一棵 n 个点、有边权的树，再给定 m 条路径，路径花费时间为其边权和。你可以将某一条边权设为 0，使得这 m 条路径同时出发，全部到达的时刻最早。

$n, m \leq 3 \times 10^5$，边权不超过 1000。

解

首先可以二分时间，然后需要删除一条边，使得所有路径都能在规定时间内完成。这要求所有超时的路径经过同一条边，并且删去这条边能够使最长的路径符合规定，路径的交可以用树上差分处理并枚举。

时间复杂度 $O((n+m)\log(nw))$。

[UOJ 261] 天天爱跑步

给定一棵 n 个点的树，另有 m 条路径 $s \to t$ 从 0 时刻出发，每过一时刻走过一条边。对于每个点有一个参数 W_j，求有 m 条路径有多少条恰好在 W_j 时刻经过该点。

$n, m \leq 3 \times 10^5$。

[UOJ 261] 天天爱跑步

给定一棵 n 个点的树，另有 m 条路径 $s \to t$ 从 0 时刻出发，每过一时刻走过一条边。对于每个点有一个参数 W_j，求有 m 条路径有多少条恰好在 W_j 时刻经过该点。

$n, m \le 3 \times 10^5$。

解

将路径 $s \to t$ 拆成 $s \to a$ 和 $a \to t$ 两部分，其中 a 为 LCA。则 $s \to a$ 经过点 j 的时间为 $w + \mathrm{dep}_s - \mathrm{dep}_j$，可以树上差分维护 $w + \mathrm{dep}_s$ 的桶；$a \to t$ 的部分同理。

时间复杂度 $O(n + m \log n)$。

树的直径

给定一棵带正边权的树，求树上最长路径。

树的直径

给定一棵带正边权的树，求树上最长路径。

解一：树上 DP

设 f_i、g_i 分别表示 i 子树内最远、次远点（不经过同一个儿子），答案即为 $f_i + g_i$ 的最大值。其中 f、g 可以树上 DP 求出。

时间复杂度 $O(n)$。

树的直径

给定一棵带正边权的树，求树上最长路径。

解一：树上 DP

设 f_i、g_i 分别表示 i 子树内最远、次远点（不经过同一个儿子），答案即为 $f_i + g_i$ 的最大值。其中 f、g 可以树上 DP 求出。

时间复杂度 $O(n)$。

解二：两次 DFS

从树上任何一个点开始 DFS 求出与其最远点，再从该点出发求出最远点，这两点之间的路径即为直径。

时间复杂度 $O(n)$。

树的直径

定理

树上距离任意一点最远的点一定是某条直径直径的端点。

树的直径

定理

树上距离任意一点最远的点一定是某条直径直径的端点。

证明.

反证法，如果存在直径的两端点距离都比该点进，则可以得到一个更长的直径。 □

树的直径

定理

树上距离任意一点最远的点一定是某条直径直径的端点。

证明.

反证法，如果存在直径的两端点距离都比该点进，则可以得到一个更长的直径。 □

定理

树上所有直径的中点重合。

树的直径

定理

树上距离任意一点最远的点一定是某条直径直径的端点。

证明.

反证法，如果存在直径的两端点距离都比该点进，则可以得到一个更长的直径。 □

定理

树上所有直径的中点重合。

证明.

反证法，如果有两条直径的中点不相同，则连接这两个中点可以得到一个更长的直径。 □

[BZOJ 1999] Core 树网的核

给定一棵 n 个点、带正边权的树，求一条长度不超过 s 的路径，满足其在一条直径上，且其到树上的最大距离最小。

$n \leq 5 \times 10^5$，权值小于 500。

[BZOJ 1999] Core 树网的核

给定一棵 n 个点、带正边权的树，求一条长度不超过 s 的路径，满足其在一条直径上，且其到树上的最大距离最小。

$n \leq 5 \times 10^5$，权值小于 500。

解

可以证明，任意一条直径直径上都可以找出符合条件的路径(事实上，答案一定在所有直径的交上存在)。因此我们任找出一条直径，求出其上每个点不经过直径的最大距离，然后双指针单调队列维护距离最大值即可。

时间复杂度 $O(n)$。

[Luogu P3629] 巡逻

给定一棵 n 个点的树，你需要加入 k 条边并规划一条巡逻路径，满足从某一点出发经过每条原有道路至少一次，并经过新建道路恰好一次，求路径长度最小值。

$n \le 10^5$，$k \le 2$。

[Luogu P3629] 巡逻

给定一棵 n 个点的树，你需要加入 k 条边并规划一条巡逻路径，满足从某一点出发经过每条原有道路至少一次，并经过新建道路恰好一次，求路径长度最小值。

$n \leq 10^5$，$k \leq 2$。

解

当 $k = 0$ 时，每条边都需要被经过两次，答案即为 $2(n-1)$。

[Luogu P3629] 巡逻

给定一棵 n 个点的树，你需要加入 k 条边并规划一条巡逻路径，满足从某一点出发经过每条原有道路至少一次，并经过新建道路恰好一次，求路径长度最小值。

$n \le 10^5$，$k \le 2$。

解

当 $k = 0$ 时，每条边都需要被经过两次，答案即为 $2(n-1)$。

当 $k = 1$ 时，新增一条边形成一个环，环上的边只需要经过一次，为了节约的长度最多，应该连接直径两端点。

[Luogu P3629] 巡逻

给定一棵 n 个点的树，你需要加入 k 条边并规划一条巡逻路径，满足从某一点出发经过每条原有道路至少一次，并经过新建道路恰好一次，求路径长度最小值。

$n \le 10^5$，$k \le 2$。

解

当 $k = 0$ 时，每条边都需要被经过两次，答案即为 $2(n-1)$。

当 $k = 1$ 时，新增一条边形成一个环，环上的边只需要经过一次，为了节约的长度最多，应该连接直径两端点。

当 $k = 2$ 时，再新增的环可能与之前的环重叠，重叠部分需要经过两次为负收益，因此把之前的直径上的边权改为 -1 再连接新直径两端点即可。

时间复杂度 $O(n)$。

[UOJ 347] 通道

给三棵 n 个点的数 T_1、T_2、T_3，求:

$$\max_{u,v}\{\mathrm{dis}_1(u,v)+\mathrm{dis}_2(u,v)+\mathrm{dis}_3(u,v)\}$$

$n \le 10^5$，边权在 $[0,10^{12}]$ 内。

[UOJ 347] 通道

给三棵 n 个点的数 T_1、T_2、T_3，求:

$$\max_{u,v}\{\mathrm{dis}_1(u,v)+\mathrm{dis}_2(u,v)+\mathrm{dis}_3(u,v)\}$$

$n \le 10^5$，边权在 $[0,10^{12}]$ 内。

只需考虑一棵树、一棵树 $+$ 一条链、一棵树 $+$ 两条链、两棵树的情况。

[UOJ 347] 通道

给三棵 n 个点的数 T_1、T_2、T_3，求：

$$\max_{u,v}\{\operatorname{dis}_1(u,v)+\operatorname{dis}_2(u,v)+\operatorname{dis}_3(u,v)\}$$

$n \le 10^5$，边权在 $[0,10^{12}]$ 内。

只需考虑一棵树、一棵树 + 一条链、一棵树 + 两条链、两棵树的情况。

解

性质：两个点集合并后的直径端点一定是原来的直径端点。

[UOJ 347] 通道

给三棵 n 个点的数 T_1、T_2、T_3，求：

$$\max_{u,v}\{\text{dis}_1(u,v)+\text{dis}_2(u,v)+\text{dis}_3(u,v)\}$$

$n\le 10^5$，边权在 $[0,10^{12}]$ 内。

只需考虑一棵树、一棵树 $+$ 一条链、一棵树 $+$ 两条链、两棵树的情况。

解

性质：两个点集合并后的直径端点一定是原来的直径端点。

因此在一棵树上枚举 LCA，维护子树内点集在另一棵树上的直径端点，额外的距离为 $\text{dep}_u+\text{dep}_v-2\text{dep}_{\text{LCA}}$，其中 LCA 已知，将 u、v 的深度挂在第二棵树上即可。

时间复杂度 $O(n\log n)$ 或 $O(n)$。

树的重心

- 所以子树大小不超过树大小的一半；
- 最大子树的大小最小。

树的重心

- 所以子树大小不超过树大小的一半；
- 最大子树的大小最小。

解

直接 DFS 求出子树大小，枚举点判断即可。
时间复杂度 $O(n)$。

树的重心

- 所以子树大小不超过树大小的一半；
- 最大子树的大小最小。

解

直接 DFS 求出子树大小，枚举点判断即可。
时间复杂度 $O(n)$。

性质

- 树的重心到所有点的距离和最小；

树的重心

- 所以子树大小不超过树大小的一半；
- 最大子树的大小最小。

解

直接 DFS 求出子树大小，枚举点判断即可。
时间复杂度 $O(n)$。

性质

- 树的重心到所有点的距离和最小；
- 合并两棵树时，新树的重心在原来两个重心间的路径上；

树的重心

- 所以子树大小不超过树大小的一半；
- 最大子树的大小最小。

解

直接 DFS 求出子树大小，枚举点判断即可。
时间复杂度 $O(n)$。

性质

- 树的重心到所有点的距离和最小；
- 合并两棵树时，新树的重心在原来两个重心间的路径上；
- 树的重心只有一个或两个（即重心在一条边的中间）。

[BZOJ 3302] 树的双中心

给定一棵 n 个点的树，带点权 W_i，请选出两个点 x、y，使得下式最小：

$$\sum_{v=1}^{n} W_v \cdot \min\{\mathrm{dis}(v, x), \mathrm{dis}(v, y)\}$$

$n \le 50000$，深度不超过 100。

[BZOJ 3302] 树的双中心

给定一棵 n 个点的树，带点权 W_i，请选出两个点 x、y，使得下式最小：

$$\sum_{v=1}^{n} W_v \cdot \min\{\mathrm{dis}(v,x), \mathrm{dis}(v,y)\}$$

$n \le 50000$，深度不超过 100。

解

选定的 x、y 对应的 v 一定由一条边分成两部分，枚举这条边断开，x、y 即为两边的重心。提前处理出子树大小，寻找重心只需要往大的子树走即可。

时间复杂度 $O(nh)$。

[Luogu P5666] 树的重心

给定一棵 n 个点的树，求断开每条边后两个连通块所有重心（可能有两个）编号之和。

不超过 5 组数据，$n \leq 3 * 10^5$。

[Luogu P5666] 树的重心

给定一棵 n 个点的树，求断开每条边后两个连通块所有重心（可能有两个）编号之和。

不超过 5 组数据，$n \leq 3 * 10^5$。

解

对于两个重心的情况，我们只需求出其中任意一个，多跳一步即可判断另一个。

先考虑断掉一条边下方整子树的部分，每次合并两棵子树时，重心一定是较大子树原重心的祖先，且移动的距离不超过小子树的大小，直接暴力向上跳即可。

再考虑上方部分，每次从最大的子树继承，将其余小子树删除，重心一定向上或向其他子树移动，其他子树此时的大小是不变的，而移动距离不超过最小子树大小，因此记录最大、次大子树后也可以直接暴力跳。

总时间复杂度 $O(n \log n)$。

树链剖分

将一棵树划分成若干条从祖先到后代链，使得每个点恰好在一条链上。

树链剖分

将一棵树划分成若干条从祖先到后代链，使得每个点恰好在一条链上。

重链剖分

每个点与其最大的儿子划分为同一条链，其余儿子划分为新链。

树链剖分

将一棵树划分成若干条从祖先到后代链，使得每个点恰好在一条链上。

重链剖分

每个点与其最大的儿子划分为同一条链，其余儿子划分为新链。

性质

在重链剖分中，每个点到根的路径至多经过 $O(\log n)$ 条轻边（及同样条重链）。

树链剖分

将一棵树划分成若干条从祖先到后代链，使得每个点恰好在一条链上。

重链剖分

每个点与其最大的儿子划分为同一条链，其余儿子划分为新链。

性质

在重链剖分中，每个点到根的路径至多经过 $O(\log n)$ 条轻边（及同样条重链）。

证明.

每经过一条轻边，子树大小至少翻倍，而子树最终大小只为 n。 □

树链剖分

重链剖分求 LCA

　　如果两点不在一条重链上，则链首深度较深的点跳出这条链；直至两点在同一条链上时，深度小的点即为答案。

　　预处理时间复杂度 $O(n)$，单次最坏时间复杂度 $O(\log n)$，常数因子优秀。

[Luogu P2146] 软件包管理器

有 n 个软件包，软件包间的依赖关系形成了一棵树（每个软件包依赖其父亲），安装一个软件包需要安装其所有依赖，卸载一个软件包将同时卸载依赖其的软件包。有 q 次安装或卸载操作，求每次操作改变状态的软件包个数。

$n, q \leq 10^5$。

[Luogu P2146] 软件包管理器

有 n 个软件包，软件包间的依赖关系形成了一棵树（每个软件包依赖其父亲），安装一个软件包需要安装其所有依赖，卸载一个软件包将同时卸载依赖其的软件包。有 q 次安装或卸载操作，求每次操作改变状态的软件包个数。

$n, q \leq 10^5$。

解

问题等价于树的子树覆盖、子树求和、到根覆盖、到根求和问题，可以直接用树链剖分和线段树解决。

时间复杂度 $O(n\log^2 n)$。

[BZOJ 3626] LCA

给定一棵 n 个点的有根树，有 q 次询问，每次给定 l、r、z，求 $[l, r]$ 中的点与 z 的 LCA 深度之和。

$n, q \leq 50000$。

[BZOJ 3626] LCA

给定一棵 n 个点的有根树，有 q 次询问，每次给定 l、r、z，求 $[l,r]$ 中的点与 z 的 LCA 深度之和。

$n, q \leq 50000$。

解

发现 LCA 的深度等于公共祖先的个数，因此只要统计所有公共祖先即可。

将询问离线差分，按照顺序加入点 $1 \ldots r$，每次加入点时对所有祖先 $+1$，查询 z 时则查询 z 的所有祖先，可以用重链剖分维护。

时间复杂度 $O(q \log^2 n)$。

[Luogu P5659] 树上的数

给定一棵 n 个点的树，每个点上都有一个球，写着 $[1,n]$ 中的某个整数，不存在两个球上的数字相同。

你可以进行 $n-1$ 次操作，每次选定一条边，交换这条边两端的球。每条边交换恰好一次。

设 p_i 表示写着 i 的球所在结点的编号，求 p 的字典序最小值。

不超过 10 组数据，$n \leq 2000$。

[Luogu P5659] 树上的数

给定一棵 n 个点的树，每个点上都有一个球，写着 $[1,n]$ 中的某个整数，不存在两个球上的数字相同。你可以进行 $n-1$ 次操作，每次选定一条边，交换这条边两端的球。每条边交换恰好一次。设 p_i 表示写着 i 的球所在结点的编号，求 p 的字典序最小值。

不超过 10 组数据，$n \le 2000$。

解

每次考虑将点 u_1 上的球通过路径 $u_1, u_2 \ldots, u_k$ 换到 u_k，则要求：

- 对于 u_1，边 (u_1, u_2) 是其选择的第一条边；
- 对于 u_i，边 (u_{i-1}, u_i) 和 (u_i, u_{i+1}) 是其连续选择的两条边；
- 丢与 u_k，边 (u_{k-1}, u_k) 是其选择的最后一条边。

[Luogu P5659] 树上的数

给定一棵 n 个点的树，每个点上都有一个球，写着 $[1, n]$ 中的某个整数，不存在两个球上的数字相同。你可以进行 $n-1$ 次操作，每次选定一条边，交换这条边两端的球。每条边交换恰好一次。设 p_i 表示写着 i 的球所在结点的编号，求 p 的字典序最小值。

不超过 10 组数据，$n \leq 2000$。

解

每次考虑将点 u_1 上的球通过路径 $u_1, u_2 \ldots, u_k$ 换到 u_k，则要求：

- 对于 u_1，边 (u_1, u_2) 是其选择的第一条边；
- 对于 u_i，边 (u_{i-1}, u_i) 和 (u_i, u_{i+1}) 是其连续选择的两条边；
- 丢与 u_k，边 (u_{k-1}, u_k) 是其选择的最后一条边。

考虑从小到大枚举球，将其换到可能的最小编号。对于每个点，边的关系可以用链表维护。时间复杂度 $O(n^2)$。