树形DP

P8625 [蓝桥杯 2015 省 B] 生命之树 满足连通性的点集和最大

我的思路过程：

考虑一个简单的结构，一个根节点，两个儿子。

如果我们希望两个儿子连通，那么根节点一定要选取。

如果根节点不选取，则两个儿子一定都不选。

由此，考虑DP，dp_u 表示以 u 为根节点，所能选取的最大值。

$$dp_u = a_u + \sum max(dp_v, 0)$$

最后的答案就是，$\max_{1-n} dp_i$

P8744 [蓝桥杯 2021 省 A] 左孩子右兄弟 根的最大高度由子树转移

首先，贪心的想，怎么取到最大高度呢？

最大高度即根节点儿子数 + 儿子中的最大高度。

转移方程：

$$dp_u = size + \max_{u->v} dp_v$$

size如果用邻接表存图，可以直接用成员方法。

P8602 [蓝桥杯 2013 省 A] 大臣的旅费

首先我们介绍以下树上路径的特点

树上两点的最短距离

$$dis(x, y) = dep[x] + dep[y] - 2dep[LCA(x, y)]$$

树上两点的简单路径，一定经过他们的最近公共祖先。

那我们如果要找当前这个树上的最长路径，考虑树形DP，应该怎么写呢

对于一个根节点，同时记录他的以它为根的最大值与次大值。

那么直径就是这里对于每个根节点，d1 + d2的值。

令 $dp[u][1]$ 是最大值， $dp[u][0]$ 是次大值

```
void DP(int u, int p)
```

```
{
    priority_queue < int > q;
    q.push(0); q.push(0);
    for (auto t : linker[u])
    {
        int v = t.fi, val = t.se;
        if (v == p) continue;
        DP(v, u);

        q.push(dp[v][0] + val);
    }
    dp[u][0] = q.top(); q.pop();
    dp[u][1] = q.top();
    return ;
}
```

OK，恭喜你，你已经会如何求 **“树的直径”** 了

以上是，树的直径的一种求法。

树的直径还有一种求法，是两遍DFS。

[C. Tree Cutting](#) 二分答案 + 树形DP（树分块基础）

考虑二分，树块的大小。

然后DP，以这个大小分块一共分了几块。验证答案。

关于树的杂项

[P8655 [蓝桥杯 2017 国 B] 发现环](#) （基环树基础）树上找环

基环树，是 n 个点，n 条边的图。

处理基环树问题，一般转化为 树 + 环 来解。

这题比较简单，无向图跑一次拓扑排序就有环了。

[P8805 [蓝桥杯 2022 国 B] 机房](#)

这题是树上前缀和

令dp_u 表示，从1到 u 的点权和

$$ans = dp_u + dp_v - 2 * dp_{LCA(u,v)} + a_{LCA(u,v)}$$

这个过程，稍微改一改，就是树上差分。

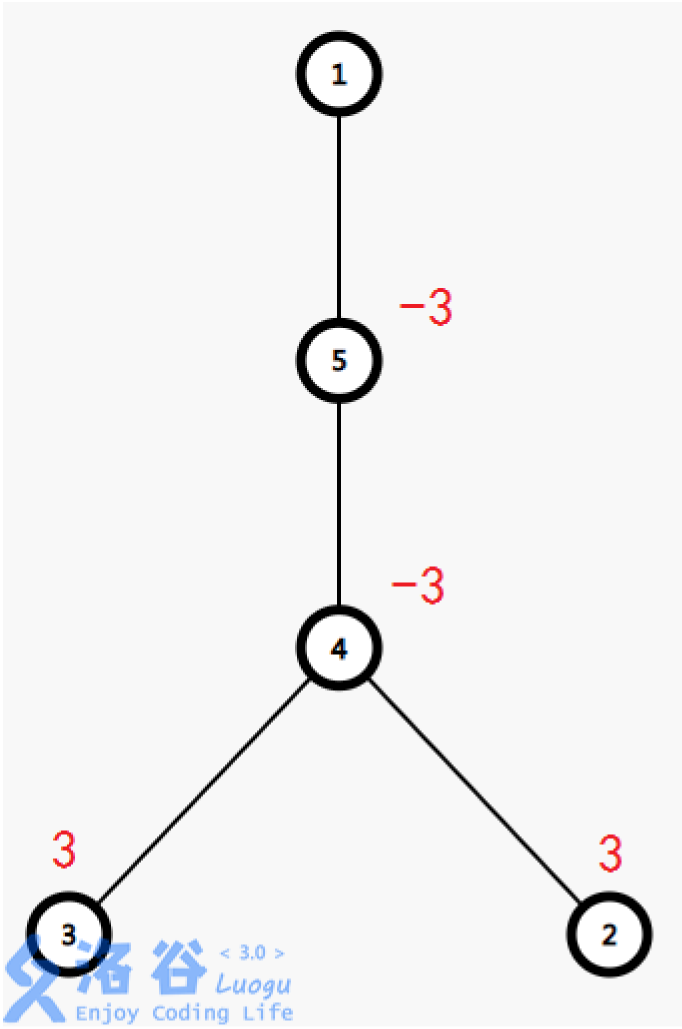
回忆以下，什么是差分？

如果题目要求对树上的一段路径进行操作，并询问某个点或某条边被更改后的权值

树上差分：树上点差分，树上边差分

点差分

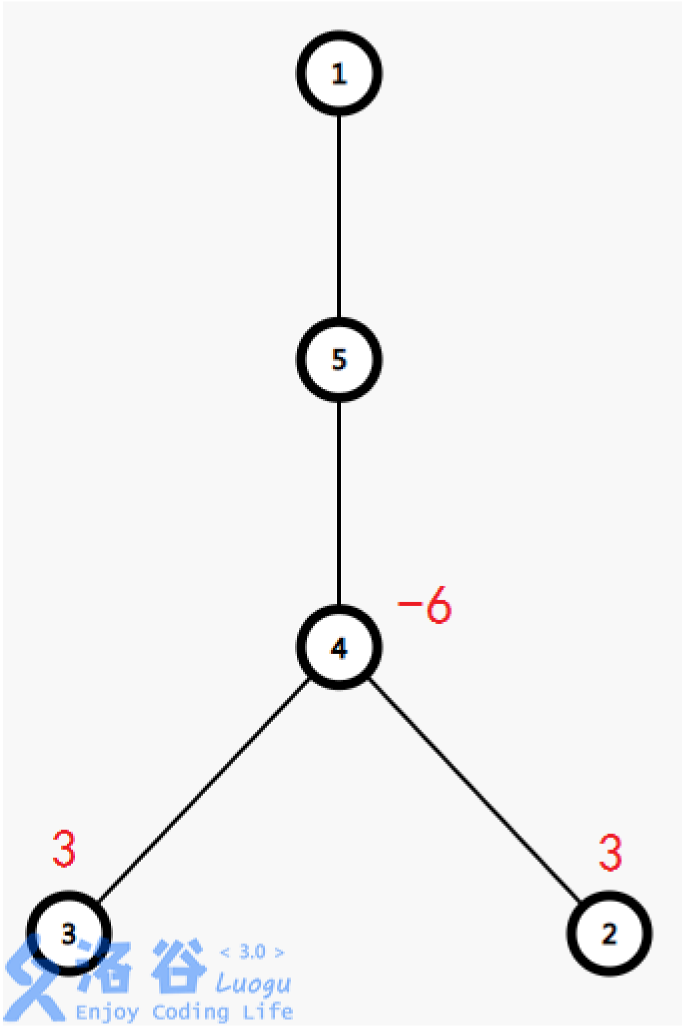
$$dif_u+1 \quad dif_v+1$$
$$dif_{lca}-1 \quad dif_{fa(lac)}-1$$



边差分

$$dif_u+1 \quad dif_v+1$$
$$dif_{lca}-2$$

自身子树和即为 $<u,fat_u>$ 的访问次数（修改后的权值）



串门 树的直径

tips : 选定一个起点，从该点出发，经过树上所有点再回来。路径长度是树边权和的两倍

树的直径求法，树形DP，或者两遍DFS。

树的直径的若干性质

定义：如果在树中选择某个节点并删除，这棵树将分为若干棵子树，统计子树节点数并记录最大值。取遍树上所有节点，使此最大值取到最小的节点被称为整个树的重心。

- 树的重心如果不唯一，则至多有两个，且这两个重心相邻。
- 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。
- 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。
- 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。
- 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。

Hard

P9235 [蓝桥杯 2023 省 A] 网络稳定性 生成树也是树，在Kruskal上求LCA