《蜀道难》解题报告

西北工业大学附属中学 刘承奥

1 题意

给一棵树标号 1 到 n，使得每条边两端差绝对值之和最小，且（存在一个点到所有点路径标号单调）。

形式化题意

对于一棵有标号有根树 $T=(V,E)$，标号 $p: v\to p(v), v\in V, p(v)\in[1,|V|]\cap\mathbb{Z}$ 是一个一一映射。令一条边 $e=(u,v), e\in E$ 的边权为：$w: e\to w(e)=|p(u)-p(v)|, e\in E, w(e)\in\mathbb{Z}$。令整棵树的权为：$W: T=(V,E)\to W(T)=\sum_{e\in E} w(e)$。

另外定义一个图 $G(T)=(V,E')$，其中 $(u,v)\in E'$ 当且仅当在 T 中 u 到 v 路径上点的标号 $p_1,p_2,\cdots,p_l$，要么单调递增，要么单调递减。则 p 必须使得 $G(T)$ 的直径不超过 2，即 $\max_{i,j\in V} SP(i,j)\le 2$，其中 $SP(i,j)$ 表示 $G(T)$ 中 i,j 的最短路经过的边数。

给定 T，求 $M(T)=\min_p W(T)$。

2 解法

引理 1 $\max_{i,j\in V} SP(i,j)\le 2$ 的充要条件是存在一个点到所有点路径都单调。

证：充分性显然。必要性：首先，假设 $(x,y),(y,z)$ 路径均单调，若 y 不在 (x,z) 上，取二者对称差可得一条同样合法的路径。由此得到原条件等价于树上任意路径可以划分成至多两个单调的串。假设一段路径需要分成两段，显然分割点是唯一的，设为 $D(i,j)$，而对于同一个 i，所有的 $D(i,j)$ 必定分布在同一以 i 为起点的路径 $P(i)$ 上。同时有 $\forall i,j, P(i)\cap P(j)\ne\emptyset$，而若干两两相交的路径一定有全体公共点，这个点到所有点都单调。

那么关注这个点（不妨设为 O）。不妨把树看作以该点为根的有根树。

显然，单调路径分单调上升和单调下降两种。事实上我们可以把这个树看作一个大根堆和一个小根堆共用堆顶。假如我们已经知道了堆的划分方法 H_g, H_s，显然有 $M(T)\ge M(H_g)+M(H_s)$。而这个下界是容易达到的，只需要在公共堆顶 O 取中间值并复制两个堆的分布即可。

考虑一个堆的情况。我们可以注意到一件事，即将树的“值域”（标号的极差）压缩必然是优的，并且压缩 d 的大小至少会减小 d 的代价。那么假设在堆的若干个子树中将某些子树进行压缩，两个相交的子树 S_1,S_2，假如 S_1 在 S_2 的值域中有 d_1 个元素，反之有 d_2 个，将两棵子树分别按顺序重排的代价至少

减少了 $d_1 + d_2$。等于重排过程中二者端点移动距离和；然而我们付出的代价只是其中一者（远离堆顶）的。这启示我们堆的每个子树值域都应该连续。

要证明这一点比较复杂，我有一些详细的归纳讨论，但在此略去。我们将结论表述如下：

引理 2 一个堆必然存在一种最优策略使得每棵子树标号值域连续。

直接使用这个引理做背包 DP 可以得到一个单次询问 $O(n^2)$ 的算法。但我们可以更详细地讨论：

有了这条引理，我们的一棵子树的代价就可以分为其各个子树内部的代价之和加上它们之间排列顺序的代价。假设子树大小分别为 $size_i$，标号从大到小顺序为 p_i，那么答案就是 $\sum_i size_i(p_i - 1) + 1$，明显地，$p$ 应该按 $size$ 的反方向顺序分配。（因为乘法满足四边形不等式）

事情到这里就比较简单了。我们还需要考虑两个堆公共顶点处，每棵子树分配给哪个堆，但稍加分析就可发现，实际上按大小顺序依次交错分给两个堆是最优的，即代价为 $\sum_i size_i \lfloor \frac{p_i - 1}{2} \rfloor + 1$。

现在，静态问题已经可以在 $O(n)$ 时间（子树排序可以使用计数方法）内解决了。

那么如何修改呢？

我们可以注意到每个点为根的答案都被算了一次，假如只要算一个点，问题就会简单很多。但遗憾的是两个堆与点度和子树大小有关的乘积缺乏大小规律（假如是一个堆，或者两个不公共顶点而是顶点以边相连的堆，那倒简单地多了。前者最优解是叶子，后者在重心。它们都避免了至少一种因素的影响。）

那么换一种思路，暴力维护每个点的答案。由于我们的决策与子树大小有关，那么我们有一个基本的问题：子树大小顺序的改变次数有多少呢？答案是有些出乎意料的：

引理 3 一棵树添加或者删除一个叶子，子树大小顺序被改变的点数为 $O(\log n)$。

我们考虑，假如原本子树就是最大的，加入叶子显然不改变大小；而次大的也很难改变——这是因为最大子树就是含有重心的子树，而重心每次至多移动 1 的距离而已。于是我们可以以重心为根的有根树上做重链剖分，显而易见地，被改变的点就只能是轻边的父亲了。

那么直接维护就可以了。这里还要注意一点：我们各式各样的贡献可以归为五种修改：单点加减、子树加减、重链加减、若干个大小顺序连续的子树加减、改变单个子树顺序。维护全局最小值。那么括号序列树应该是一个好选择。另外重心移动的细节影响也很多。令人欣慰的是，重心刚刚移动时，原来的点必然是新重心的最大子树，因而不会带来额外影响。

但实现细节非常复杂。

时间复杂度：$O(n \log n + q \log^2 n)$

空间复杂度：$O(n + q)$

3 题目背景

本题是由一个形式更简洁的问题衍生而来：不要求路径单调，只求每条边两端差绝对值之和的最小值。然而这个形式简洁的问题却要困难得多，我们连多项式时间复杂度的算法都没有找到（或查到）。

引理 2 在一般的问题中不成立，这就为状态的化简和划分造成了极大的困难。也许需要一种全新的理论才能解决。我也怀疑过它是一个 NP 难题，但它的条件和祖玛的强制消除一样，很难规约证明。

4 总结

本题考查的内容有：组合决策和组合优化：贪心算法，数据结构：树链剖分，Link-Cut Tree 等。

本题并不算一道有趣的好题。它繁琐的的实现细节远远超出设计算法的难度之上，作为一道题目而言或许练习代码能力才是它的主要价值。不过，这也是背景问题难度的反面表现。