数据结构 解题报告

朱屹帆

2024.10.09

题目大意

给定一棵 n 个节点的树，钦定点 1 为根，每条边的边权均为 1 ，节点 x 有点权 a_x ，初始所有节点的点权均为 0 。总共有 m 次操作，每次操作先给出操作类型 opt ，接下来给出该操作的所有参数。

1. 给定一条路径 (x, y) 和范围 k 以及权值 u ，对于所有点 i ，当点 i 满足其到树上路径 (x, y) 的最短距离 $\le k$ 时，有 $a_i \leftarrow a_i + u$ 。

2. 给定点 x 以及权值 u ，对于所有点 x 子树内的节点 i ，有 $a_i \leftarrow a_i + u$ 。

3. 给定一条路径 (x, y) 和范围 k ，对于所有点 i ，当点 i 满足其到树上路径 (x, y) 的最短距离 $\le k$ 时，询问所有这样的点 i 的点权和。

4. 给定点 x ，对于所有点 x 子树内的节点 i ，询问所有这样的点 i 的点权和。

5. 给定一条路径 (x, y) 和范围 k ，对于所有点 i ，当点 i 满足其到树上路径 (x, y) 的最短距离 $\le k$ 时，询问所有这样的点 i 的点权 $\max$ 。

6. 给定点 x ，对于所有点 x 子树内的节点 i ，询问所有这样的点 i 的点权 $\max$ 。

其中编号对应操作类型，操作的所有参数将以上述描述的变量顺序给出。

数据范围

本题开启子任务评测，子任务之间将会设置合理的依赖关系。

子任务 1 和子任务 2 (10 分 + 10 分): 保证 $k = 0$ 。

子任务 3 和子任务 4 (10 分 + 10 分): 保证所有涉及路径 (x, y) 的操作满足 $x = y$ 。

子任务 5 和子任务 6 (10 分 + 10 分): 保证 $k = 1$ 。

子任务 7 和子任务 8 (20 分 + 20 分): 无特殊限制。

上述子任务中所有奇数编号子任务不包含操作五和操作六。

对于所有数据，保证 $1 \le n, m \le 2 \times 10^5$，$1 \le k \le 3$，$-10^5 \le u \le 10^5$ 。

时间限制 7 s，空间限制 512 MB。

解题过程

下文中认为 n 和 m 同阶，题解部分内容可以见 参考资料 中笔者的 2023 年集训队互测《数据结构》解题报告。

算法一

当 $k=0$ 时，使用树链剖分维护链和子树的修改与查询，时间复杂度 $O(n\log^2 n)$ ，期望得分 10 到 20 分。

算法二

当保证操作一、操作三和操作五的路径 (x,y) 均满足 $x=y$ ，且不包含操作五和操作六时，考虑对操作序列进行根号重构，并分析前面的所有修改操作对后续的查询操作的影响。通过分析可以发现修改操作对查询操作的贡献即为两者操作的范围交的点的数量乘以修改操作对应权值。

操作一对操作三的贡献即为两个单点的对应邻域交的大小乘以修改权值，树上任意两个单点的邻域交必然可以表示成一个点的邻域或一条边的中点的邻域，而找到这个中点是容易的，由于 k 的数据范围较小，因此两个单点只有在树上距离 $\le 2k$ 时邻域交不为空，此时可以取出两个单点在树上的路径并找到对应中点。

操作一对操作四的贡献即为一个单点的邻域和一个单点的子树的交的大小乘以修改权值，通过类似 dp 的方式预处理单点子树内外邻域的一些信息，随后即可根据修改单点和查询单点的相对位置关系快速得到交的大小。

操作二对操作三的贡献即为一个单点的子树和一个单点的邻域的交的大小乘以修改权值，类似操作一对操作四的贡献分析。

操作二对操作四的贡献即为两个单点的子树交的大小乘以修改权值，这是简单的，根据两个单点的祖先后代关系确定子树交得到的新的子树。

可以发现上述贡献分析在根号重构时是相对容易维护的，使用类似 dp 的方式下放修改信息即可。

具体细节可以见 参考资料 中笔者的 2023 年集训队互测《数据结构》解题报告。由于查询 max 信息并不支持将修改操作拆分后类似的考虑，因此该做法似乎无法拓展到维护操作五和操作六。时间复杂度 $O(n\sqrt{n}\mathrm{poly}(k))$ ，结合算法一，期望得分 30 分。

算法三

注意到算法二的核心思想在于分析修改对询问的贡献，不妨尝试换一种思考方向，当 $k=1$ 且不存在操作五和操作六时，得到类似 $k=0$ 的处理方法。

简单回顾算法一，当 $k=0$ 时，根据轻重链剖分的性质，从任意一个点开始不断跳到父节点，直至根节点时停止，在这个过程中至多会跳 $\log n$ 条轻边，因此可以将一条路径拆分成 $\log n$ 条重链，并且由于在树链剖分的 dfs 序中，重链上的节点的标号具有连续性，因此可以使用数据结构维护。这启发我们可以尝试在保持轻重链剖分对 $\log n$ 条重链进行处理以代替对一条路径进行处理的性质的基础上，通过调整重链的标号方式，使得重链邻域节点的标号大致连续。

笔者将该思想称为“基于调整搜索顺序的重标号树链剖分”，具体细节可以见 参考资料 中笔者 2024 年的集训队论文。接下来将介绍一种能够处理上述模型的标号方式。

首先采用传统树链剖分的标号方式中的顺序考虑每一条重链。对于一条重链，首先将该重链中的所有点按照深度从小到大依次进行标号，接下来仍然以深度从小往大的顺序扫描该重链的所有节点，对于当前扫描的节点 x ，将点 x 的所有轻儿子进行标号。需要注意的是由于该标号方式会对一条重链及其一度邻域进行标号，因此可能会出现在标号过程中一个节点已经被标号的情况，此时不改变该节点的原有标号。接下来将会说明该标号方式的部分性质以及基于该标号方式的维护方式。

可以发现该标号方式仍然保持了树链剖分的许多优秀性质，例如可以将一条路径大致定位到 $\log n$ 条重链上进行处理，更为关键的是该标号方式使得树的一度邻域信息在标号中具有了一些连续性，且标号过程中出现节点已经被标号的情况仅有可能在一条重链的链顶发生，因为对点 x 的轻儿子进行标号的过程仅会影响到其他重链的链顶节点。

对于操作一和操作三，取出一条重链上点 x 到点 y 的路径一度邻域信息，不妨有点 x 为点 y 祖先节点，则一度邻域路径即为点 x 一度邻域最小标号到点 y 一度邻域最大标号构成的标号区间，此时由于标号的特殊的连续性，因此可以使用数据结构进行维护。对于操作二和操作四，注意到上述标号方式仍然保证了类似子树的标号性质，因此将子树对应标号区间取出后类似的数据结构维护即可。上述分析中由于存在重链链顶节点标号不与重链其他节点保持连续性，因此需要补充重链链顶以及重链区间末尾端点的重儿子的贡献，这是简单的细节部分。

具体细节可以见 参考资料 中笔者的 2023 年集训队互测《数据结构》解题报告。时间复杂度 $O(n\log^2 n)$ ，结合算法一，期望得分 30 分。

算法四

可以发现算法三似乎具有极大的拓展空间，不妨尝试将其拓展到更加经典的模型中，当不存在操作五和操作六时，得到足够优秀的处理方法。考虑进一步优化标号方式，不妨设 $a=3$.

采用传统树链剖分的标号方式中的顺序考虑每一条重链。对于一条重链，首先将该重链中的所有点按照深度从小到大依次进行标号，接下来循环以下过程 a 轮：对于第 i 轮，以深度从小往大的顺序扫描该重链的所有节点，对于当前扫描的节点 x ，将点 x 的

所有轻儿子子树内距离点 x 恰好为 i 的所有节点进行标号。值得注意的是该标号方式会对一条重链及其 a 度邻域内的所有点进行标号，因此可能会出现在标号过程中一个节点已经被标号的情况，此时不改变该节点的原有标号。接下来将会说明基于该标号方式的维护方式。

对于四种类型的操作，大致的重链邻域信息以及子树信息的维护是相对容易的。但可以发现，此时一条重链顶端的 a 层内的所有节点可能会和重链的点的标号不相关，而这 a 层涉及到的标号不相关的节点数量并没有理论限制，因此接下来将分析重链顶端这 a 层的点的信息所具备的优良性质。

定义 1： 对于树上的节点 x ，维护线段集合 $s0\{x,k\}$ 表示与点 x 距离恰好等于 k 的、在点 x 子树内的所有节点构成的标号区间（当 $k>a$ 时描述与点 x 的所有距离 $>a$ 的点的信息)；

定义 2： 对于树上的节点 x ，维护线段集合 $s1\{x,k\}$ 表示与点 x 距离恰好等于 k 的、在点 x 的轻儿子的子树内的所有节点构成的标号区间（当 $k>a$ 时描述与点 x 的所有距离 $>a$ 的点的信息)。

可以发现维护出具体的 $s0\{x,k\}(0\le k\le a+1)$ 和 $s_1\{x,k\}(0\le k\le a+1)$ 是容易的，只需要在完成标号后类似 dp 的方式转移处理即可。接下来称标号区间为线段。

结论 1： 对于树上任意节点 x ，可以发现 $s1\{x,k\}$ 均满足线段数量是 $O(1)$ 的，具体来说满足线段数量 ≤ 2 。

证明 1： 对于任意节点 x 而言，当 $0\le k\le a$ 时，根据上述标号方式，与其距离恰好为 k 的轻儿子节点必然会在同一时刻被连续标号，而点 x 子树内重链上距离恰好为 k 的点至多只有一个，该节点可能会将连续标号划分成两段（且简单分析可以发现只有点 x 的重儿子可能发生这种情况），因此线段数量 ≤ 2 ； 而当 $k=a+1$ 时，显然标号是完全连续的，因此线段数量 ≤ 1 。

结论 2： 对于树上任意节点 x ，可以发现 $s0\{x,k\}$ 均满足线段数量是 $O(a)$ 的，具体来说满足线段数量 $\le a+1$ 。

证明 2： 对于任意节点 x 而言，当 $0\le k\le a$ 时，节点 x 子树内的与其距离恰好为 k 的所有节点可以被表示成节点 x 及其重链对应位置的若干后继节点的 $s1\{x,k'\}$ 的并集，则可以得到 $O(a)$ 的上界，且进一步分析可以发现对于重儿子来说，每一层均满足标号的连续性，因此即使每一层的标号之间都是独立的，也仍然满足线段数量 $\le a+1$ 。而当 $k=a+1$ 时，重链的标号会将轻儿子子树的连续标号划分，因此线段数量 ≤ 2 。

因此修改单点的轻儿子子树范围内的距离 $\le k$ 的点时，线段数量为 $O(a)$ ； 修改单点的子树范围内的距离 $\le k$ 的点时，线段数量为 $O(a^2)$ 。

对于操作一和操作三，取出一条重链上点 x 到点 y 的路径 k 度邻域信息，不妨有点 x 为点 y 祖先节点，则 k 度邻域路径即为点 x 的 k 度邻域最小标号到点 y 的 k 度邻域最大标号构成的标号区间，此时由于标号的特殊的连续性，因此可以使用数据结构维护。

对于操作二和操作四，注意到上述标号方式仍然保持了类似子树的标号性质，取出子树对应标号区间后类似的数据结构维护即可。由于一条重链的前 a 层的标号与重链本身的标号并不相关，因此需要对于前 a 层进行特殊处理，不妨根据预处理得到的 $s0\{x,k\}$ 和 $s1\{x,k\}$ 信息，即在处理一条重链时，对重链区间右端点的重儿子子树以及对重链顶端的 k 层邻域进行处理。需要注意的是当涉及到处理路径 (x,y) 时，对于路径的最近公共祖先之上的部分，需要使用类似容斥的技巧补充贡献。子树的前 a 层的点的标号与子树内的点的标号并不相关，因此处理子树时需要对于前 a 层单独处理，类似的利用 $s0\{x,k\}$ 信息取出前 a 层的标号区间即可。

时间复杂度 $O(\text{poly}(a)\times n\log^2 n)$ ，上述做法大致可以做到 $O(a^2\times n\log^2 n)$ ，精细实现可以做到 $O(a\times n\log^2 n)$，常数较大，结合算法一，期望得分 60 分。

算法五

可以发现算法四似乎仍然具备极大的拓展空间，在算法四中，特殊的标号方式保持着每个点对应唯一标号的优秀性质，即点权修改操作并没有进行拆分，而是以统一的整体的形式作用到每一个点上。但算法四并不能直接查询路径 (x,y) 的 max 信息，因为在查询过程中采用了类似容斥的技巧补充贡献，而 max 信息不支持简单形式的容斥得到信息。

不妨将操作五类似操作的取出路径查询的所有区间及其贡献系数，加和信息可以直接将所有区间和乘以对应贡献系数加和得到，但查询 max 信息需要规避容斥的形式。考虑扫描线得到所有产生有效贡献的区间，将所有区间差分并在左端点加入其贡献系数，右端点后继减去其贡献系数，则扫描线过程中所有贡献系数为正数的位置即为产生有效贡献的所有位置，这是可以扫描线轻松维护的。

得到产生有效贡献的所有区间后，此时由于每个点对应唯一标号，因此查询所有区间 max 信息即可。操作六对于子树本身取出的区间并不需要容斥，直接查询即可。

时间复杂度 $O(\text{poly}(a)\times n\log^2 n)$ ，上述做法大致可以做到 $O(a^2\log a\times n\log^2 n)$ ，精细实现可以做到 $O(a^2\times n\log^2 n)$ ，常数较大，期望得分 100 分。

参考资料

笔者的 2023 年集训队互测《数据结构》解题报告。

笔者的 2024 年集训队论文《浅谈一类树上范围相关问题》。