Codechef FEB 13 Observing the Tree 解题报告

金策

September 23, 2015

1 题意简述

维护 n 个点的无根树上的点权，支持三种操作：

- c x y a b 给结点 x 到 y 路径加等差数列，首项为 a，公差为 b；
- q x y 询问结点 x 到 y 路径上的点权和；
- l x 回到第 x 次修改操作之后的状态。

操作数量为 m。强制在线。

数据规模：$1 \le n, m \le 10^5, 0 \le a, b \le 1000$。

2 题目解答

这是一道码农数据结构题。分三个部分介绍算法。

2.1 不带可持久化的链上加等差数列

先来考虑最弱化的版本：不带可持久化的链上加等差数列。

一个经典的区间修改区间查询问题，当然是喜闻乐见的线段树上打标记了。考虑一下标记怎么打。

对线段树上的每个结点 $[l_i, r_i]$，除了维护 sum_i 之外，还维护标记 a_i, d_i。表示对这个区间已经加过了一个首项为 a_i，公差为 d_i 的等差数列。

给 $[l, r]$ 加一个首项 a，公差 d 的等差数列，相当于给 $[l, mid]$ 加一个首项 a，公差 d 的等差数列，再给 $[mid + 1, r]$ 加一个首项 $a + (mid - l)d$，公差 d 的等差数列。所以线段树修改操作时是可以递归下去的。

容易看出这个标记是可以合并的，只要将 a_i, d_i 分别相加就可以了。

另外，给一个区间加上等差数列时，也很容易就能算出这个区间的 sum_i 的增加量。

所以这个问题就可以用线段树来解决了。

2.2 带可持久化的链上加等差数列

接下来我们要给这棵线段树可持久化。可持久化的方法也是人人皆知的。每当修改这个结点时，不在原来的结点上修改，而是将结点复制一份后在修改。由于一次操作只会影响 $O(\log n)$ 个结点，因此新增的结点数目也是 $O(\log n)$。

对于需要下传标记的情形，不仅要将自己复制一遍，还需要将两个儿子结点也复制一遍。这样就带了 3 倍的常数。

另一种小常数的方法是利用标记永久化。由于这里的操作是交换且结合的，因此可以使用标记永久化的手法。这样就无需下传标记，减小了常数。

2.3 带可持久化的树上加等差数列

现在要把链上的问题改为树上，于是要用树链剖分。树链剖分的方法也是人人皆知的。让每个点连一条重边连向 $size$ 最大的儿子。所有重边就组成若干条重链。用简单的手法可以证明树上的一条路径至多经过 $O(\log n)$ 条重链。因此我们对每个重链用线段树维护，那么一次操作就可以转化为 $O(\log n)$ 次重链上的线段树操作。

2.4 时空复杂度

修改操作：$O(\log^2 n)$。

询问操作：$O(\log^2 n)$。

回溯操作：$O(1)$，这只要改一下当前的版本号就可以了。

空间复杂度是 $O(n\log^2 n)$。