

《小 H 分糖果》解题报告

金天

南京市金陵中学河西分校

一、 题目大意

小 H 居住的国度形成一棵树。树是 n 个点 $n - 1$ 条边的连通图，每一个节点上是一个城市，编号为 i 的城市内居住着一位小朋友，这位小朋友期待在万圣节收到 a_i 颗糖果。

作为万圣老人，小 H 将计划他的万圣节行程，一种可能的行程是携带 m 个糖果从编号为 u 的城市出发，通过唯一的最短路走向 v 号城市，并给予居住在这条路径上（包括 u, v ）的所有小朋友一些糖果（可能某些小朋友没有收到糖果）。

在小 H 分发糖果之后，所有的小朋友（包括路径之外的）将会把收到的糖果数量 a'_i 与自己期待的糖果数量 a_i 进行比较，称一位小朋友的悲哀度为

$$\max\{0, a_i - a'_i\}^2$$

那么小 H 希望最小化所有小朋友的悲哀度之和。

现在按照时间顺序，依次发生 q 个事件。每个事件有以下两种可能：

- **1 u b**: 居住在 u 号节点的小朋友改变了主意，将他期待收到的糖果数量变为 b ，也即将 a_i 赋值为 b 。注意这个过程会影响后续的事件。
- **2 u v m**: 小 H 模拟一个万圣节行程计划，请你告诉他最小可能的悲哀度总和是多少。注意这个过程不会影响后续的事件。

二、 数据范围

对于所有数据, $1 \leq n, q \leq 10^5$, $0 \leq a_i, b \leq 10^9$, $0 \leq m \leq 10^{14}$, $1 \leq u, v \leq n$ 。

子任务 1 (10%): $n, q \leq 1000, m \leq 1000$ 。

子任务 2 (15%): $n, q \leq 1000$ 。

子任务 3 (15%): 树的形态是一条链, 即树中每个节点的度数 ≤ 2 。

子任务 4 (60%): 无特殊限制。

时间限制: 3s。

空间限制: 512 MB。

三、 解题过程

3.1 算法一

一个基础的观察是, 没有收到糖果的小朋友 i 的悲哀度显然是 a_i^2 , 那么只需要求出路径之外所有小朋友当前权值的平方和即可, 可以通过先求出路径上所有小朋友的权值平方和, 然后用所有小朋友的权值平方和减去它。

现在我们只关心路径上的小朋友, 不妨考虑一个仅由路径上小朋友的需求值组成的序列 b_1, b_2, \dots, b_k 。

考虑将 m 颗糖果逐颗分配到小朋友手中。若所有小朋友的需求均降为 0, 那么可以结束过程不再继续分配。那么一个显然的贪心方法是, 每次将糖果分配到现在需求最大的小朋友手中。由于最终悲哀度是由所有小朋友需求的平方和得到, 那么对于任意一个小朋友, 在 ta 身上分配一个糖果的价值, 随着 ta 的需求值的减少, 是一个单调递减的函数。这样在每个小朋友上分配若干糖果的总价值是一个上凸的函数, 它们的背包, 也就是凸函数闵可夫斯基和, 显然指示了一个每次把糖果分配给需求最大的小朋友的贪心方法。

那么一个基本的实现是, 对于每一颗糖果, 通过优先队列找出当前需求值最大的小朋友, 然后分配给 ta 一颗糖果。最后直接统计答案即可。时间复杂度 $O(nmq \log n)$, 可以通过子任务 1。

3.2 算法二

现在观察在分配的最后，需求值的序列形态如何。由于进行的是每次将最大值 -1 的过程，那么肯定存在若干最大值 x 以及若干 $x - 1$ ，还有一些小于 x 的没有参与分配的数。

那么如果能够找到这个最大值 x ，就可以算出将所有数减少到最大值及以下所花费的最小糖果数目，将其与 m 进行比较，多余的糖果全部用于将一部分剩余的 x 减去 1，那么就可以直接求出最终的结果。

找出这个最大值的过程非常简单，只需要找到最小的，将所有数减少到这个值及以下的糖果数不超过 m 的 x ，即为最大值。一个基本的实现是直接将所有需求值得序列排序，然后首先在序列上进行二分，求出 x 的大致范围，然后在整数上进行第二轮二分直接求出答案。时间复杂度 $O(q(n \log n + \log m))$ ，可以通过子任务 1, 2。

3.3 算法三

根据二分的想法，现在考虑如何解决链上的问题。现在需要实现的功能如下：

- 单点修改
- 进行对于 x 的二分过程，查询在区间中， $\leq x$ 的数的个数以及它们的和。

自然的想法是对值建立线段树套平衡树，外层是数值区间，内层是满足这个数值区间的下标集合。这样只需在线段树上进行二分，即可在 $O(n \log^2 n + q(\log^2 n + \log m))$ 的时间复杂度内解决。该算法可以通过子任务 3，结合算法二可以通过子任务 1, 2。

3.4 算法四

树链上的可减信息计数问题一般转化为到根信息进行处理。那么现在需要实现以下功能：

- 单点修改
- 进行对于 x 的二分过程，查询在点 u 到根的路径上， $\leq x$ 的数的个数以及它们的和。

在单点修改的过程中维护到根路径可减信息的方式是通过 dfs 序转化为区间修改，这里的“信息”指的是数集，由压缩 trie（见参考资料）维护，以保证合理的空间复杂度。

这样建立一棵以数集为基本元素的，支持区间修改与单点查询的，标记（数集）永久化的线段树。一次修改中将一个区间打上标记，一次查询中只需要找出所有用到的 trie，然后随着二分过程逐步向下走即可。

时间复杂度 $O((n + q) \log m \log n)$ ，空间复杂度 $O((n + q) \log n)$ ，通过垃圾回收可做到 $O(n \log n)$ 的空间复杂度。

可以通过子任务 1, 2, 3, 4。

四、 参考资料

- [1] Radix tree - Wikipedia, https://en.wikipedia.org/wiki/Radix_tree