

だから、もう初めて幸せになって下さい

emerald

朱羿恺

12 / 02
2023



吐槽时间

だから、もう諦めて幸せになって下さい

做法 1

- 如果 $q = 1$, 可以看做是每个点带有一个点权, 每条边带有一个边权, 需要选择一个以 x 为根的连通块使得点权之和减边权之和最大。
- 这个问题直接简单 DP 即可, 令 f_i 表示选以 i 为根的连通块的答案是多少。
- 如果对于每次询问都去 $O(n)$ DP 一遍, 可以得到 $O(nq)$ 的做法, 可以通过 Sub 1。
- 预计得分: 20

だから、もう諦めて卒業してしまおう

做法 2

- 如果可以离线，考虑对询问按照 y 从小到大排序后处理。
- 考虑对于一条边来说，如果在某个时刻被选中了，那么在之后它会一直被选中。对于点来说同理。
- 于是只需要考虑每个点什么时候会被合并到父亲。
- 考虑每个点的 f_i 形如 $ky - b$ ，则 i 和 fa_i 会在 $\frac{b}{k-w_i} > y$ 时合并，于是我们拿一个 set 维护所有点的合并时刻，每次取出最早的合并就行。
- 而对于合并的影响，相当于是一条链的 k 和 b 分别加上一个值，然后更新链顶的更新时刻即可。
- 对于 Sub 2 来说，只需要记录每段时间 1 号点 k, b ；对于 Sub 3，可以用可持久化线段树来记录每个点在每个时刻的 k, b ；对于 Sub 4，则需要树剖+可持久化线段树来支持在线查询，空间复杂度为 $O(n \log^2 n)$ ，无法通过。
- 预计得分：75

做法 3

- ▶ 来考虑一个新的想法!
- ▶ 令 $f_{i,j}$ 表示在 i 子树内买到 j 个绿宝石的最小花费。
- ▶ 可以发现只需要保留 $f_{i,j}$ 凸包上的点!
- ▶ 于是合并子树就是闵可夫斯基和，直接归并就行。
- ▶ 转移到父亲就是对凸包进行一个类似 `pop_front` 的操作。
- ▶ 询问在凸包上二分即可。
- ▶ 可以使用平衡树维护。
- ▶ 朴素实现时间复杂度为 $O(n \log^2 n + q \log n)$
- ▶ 可以使用 Splay 启发式合并的 trick 把复杂度降到 $O((n + q) \log n)$
- ▶ 预计得分：100

だから、もう初めて幸せになって下さい

Good Luck

12 / 02
2023