前往何方

ix35

Rainbow

前往何方

题目大意：

交互题。猜一棵 n 个点的树，你有一个点集 S 支持加点、删点，每次操作后交互库返回 S 导出子图最大连通块大小，最多可以进行 Q 次操作。

数据范围：

$n \leq 10{,}000,\ Q \geq 600{,}000$

前往何方 - 题解 - 算法一

询问所有点对之间有没有边。

操作复杂度为 $O(n^2)$。

10 分。

前往何方 - 题解 - 算法二

对于每个点 i 询问一次 $[n]\backslash\{i\}$ 的答案，就可以得到删掉 i 后最大连通块的大小。于是我们可以借此找到重心，并得到以重心为根时每个点 v 的子树大小 s_v。下面考虑点分治。

令 u 是重心的最大的儿子，初始集合为除重心外所有点，依次尝试删掉所有点，保持返回值为 s_u，最后我们就会得到重心的一个大小为 s_u 的子树。

前往何方 - 题解 - 算法二

将不在重子树中的点按照 s 从大到小排序，依次删除，删除后答案变小的就是根的儿子。

下面考虑怎么判定 w 是不是 v 的后代：先从全集中删除 v，再删除 w，如果答案不变则 w 是 v 的后代，否则不是。

这个方法事实上可以给定一个子集 V 判断 w 是不是 V 中任何一个点的后代。

前往何方 - 题解 - 算法二

将根的儿子二进制分组，将其中一半删掉，然后依次尝试删除其他点，就可以得出其他每个点属于哪一组。

于是做 $\log d$ 轮就可以得到每个子树的点集，递归即可。

操作复杂度 $O(n \log^2 n)$，尽量卡掉了。

前往何方 - 题解 - 算法三

我们发现，并不需要把每个子树都分开才能递归下去。事实上我们可以把所有子树分成尽量平均的两半，然后分出这两半即可。

思路 1：除掉最大子树外，剩下的子树一定可以分成两半，使得两边大小都不超过 $\frac{n}{2}$。

这种做法相当于算法二中二进制分组只分一位，操作复杂度 $O(n \log n)$，常数还是较大的。

前往何方 - 题解 - 算法三

为什么我们一定要先求出最大子树?

因为在算法二中，求其他子树的根，以及将点分进其他子树的过程中，任何时候需要保证包含重心的连通块是最大的，这可以通过保留根的最大子树做到。

我们可以换一个操作方式避免掉这个问题，从而最大子树和其他子树可以同等看待。

前往何方 - 题解 - 算法三

思路二：直接按 s 从大到小尝试删掉所有非重心结点，直到包含重心的连通块大小不到一半。这之前删掉的子树根为一边（称为左边），这之后删掉的子树根为另一边（右边）。

我们保留所有右边的子树根，删掉所有左边的子树根，这样就可以直接区分剩下的点属于左边还是右边。

操作复杂度 $O(n \log n)$，可获得 100 分。

前往何方 - 分析 - 算法三

引理：设重心的儿子大小为 $s_1 > \cdots > s_k$，若 t 使得 $s_1 + \cdots + s_t$ 与 $\frac{n-1}{2}$ 最接近，则

$$\left| s_1 + \cdots + s_t - \frac{n-1}{2} \right| \leq \frac{n-1}{6}.$$

证明：若 $s_1 \geq \frac{n-1}{3}$，则直接将 s_1 分为一边是一种可行方案，所以不等式成立。

若 $s_1 < \frac{n-1}{3}$，则 $s_t, s_{t+1} \leq s_1 < \frac{n-1}{3}$，所以假设不等式不成立，一定可以调整一下让它成立。

前往何方 - 分析 - 算法三

定义 $T(n)$ 为：初始集为全集，经过多少次操作可以求解一棵大小为 n 的树，并且退出时集合变为空集。我们有

$$T(n) \leq \begin{cases} 1 & (n = 1) \\ 2 & (n = 2) \\ \max\limits_{\frac{n-1}{3} \leq m \leq \frac{2(n-1)}{3}} \{T(m+1) + T(n-m) + 4n - 1\} & (n > 2) \end{cases}$$

算法三的操作次数为 $n + T(n)$，经过计算不超过 $600{,}000$。

前往何方 - 吐槽