Lovely Dogs 命题报告

四川省成都市第七中学 蔡欣然

2021 年 9 月

1 题目介绍

1.1 题目来源

原创题。

1.2 题目大意

给定一棵大小为 n 的树，1 号节点是树根，树上每个点有一个权值 a_i。

给定常数 d。我们将 z 拆解成一些质数的幂次的乘积 $z=\prod_i p_i^{k_i}$，我们定义：

$$f_d(z)=\prod_i (-1)^{k_i}[k_i \le d]$$

对于树上每个节点 x，你需要求出 i,j 均在 x 的子树中的所有点对 (i,j) 的 $f_d(a_ia_j)$ 的和。

注意此题中 (i,j) 和 (j,i) 是一样的点对。

1.3 时空限制

时间限制 1s，空间限制 256MB。

1.4 数据范围

对于 100% 的数据满足 $1 \le n \le 2\times 10^5, 1 \le d \le 20$，保证所有的 a_i 构成一个 1 到 n 的排列。

子任务编号	子任务分值	特殊数据范围
subtask 1	10	$n \le 500$
subtask 2	10	$n \le 2000$
subtask 3	10	$d = 20$
subtask 4	20	$d=1,\forall i,u_i=1,v_i=i+1$
subtask 5	15	$\forall i,u_i=1,v_i=i+1$
subtask 6	10	$n \le 50000$
subtask 7	25	$n \le 2\times 10^5$

2 解题过程

2.1 subtask 1

先考虑如何计算 $f_d(z)$，把 z 拆成若干质数的幂次的乘积后，按照定义式进行计算即可。由于所有需要求出 $f_d(z)$ 的 z 都是两个 $\le n$ 的数的乘积，故只用筛出 $[1,n]$ 内的素数后，依次枚举即可 $O(n)$ 计算一个 $f_d(z)$ 的值。然后先对所有 $O(n^2)$ 对 (i,j) 计算 $f_d(a_ia_j)$，时间复杂度 $O(n^3)$。

关于计算 x 子树内的答案，直接 $O(n^2)$ 枚举子树内的每一对 (i,j) 然后用先前计算出的 $f_d(a_ia_j)$ 即可，时间复杂度 $O(n^3)$。

总时间复杂度 $O(n^3)$。

2.2 subtask 2

在 subtask 1 的基础上，先把每个数 $i \in [1,n]$ 拆成若干质数的幂次的乘积，这样每对 (i,j) 计算 $f_d(a_ia_j)$ 时就可以只用管 a_i, a_j 各自拆出来的那些质数，时间复杂度 $O(n^2 \log n)$。

计算 x 子树内的答案时，把他儿子的答案都加起来，然后再加上所有以 x 为 LCA 的点对 (i,j) 的 $f_d(a_ia_j)$ 即可。时间复杂度 $O(n^2)$。

总时间复杂度 $O(n^2 \log n)$。

2.3 subtask 3

定义：

$$g(z) = \prod_i (-1)^{k_i}, z = \prod_i p_i^{k_i}$$

容易发现当 $d = 20$ 时，没有任何一个 $\le n$ 的质数的 $d+1$ 次方小于等于 n，于是此时有 $f_d(z) = g(z)$，即一个数拆解成质数的幂次的乘积后，幂次和的奇偶性。而 $g(z)$ 是一个完全积性函数，于是此时有 $f_d(a_ia_j) = g(a_i)g(a_j)$。

将树转成 DFS 序，于是求一个子树的答案相当于是求序列上的一个区间的答案，在这个区间内的所有点对 (i,j) 的 $f_d(a_ia_j)$ 的值的和。故一个区间的答案为：

$$\begin{aligned} ans &= \sum_{i=l}^{r} \sum_{j=i}^{r} g(a_i)g(a_j) \\ &= \frac{\left(\sum_{i=l}^{r} g(a_i)\right)^2 + \sum_{i=l}^{r} g(a_i)^2}{2} \end{aligned}$$

用线性筛 $O(n)$ 筛出 $g(i), i \in [1,n]$ 后，可以维护前缀和快速计算答案。时间复杂度 $O(n)$。

2.4 subtask 4

$u_i=1,v_i=i+1$ 表示这棵树是一棵以 1 为根的菊花图。对于 $x,x\geq 2$ 的点的答案为 $f_d(a_x^2)$，可以快速计算。剩下只用计算 1 的答案，即所有点对 (i,j) 的 $f_d(ij)$ 的和。

而当 $d=1$ 时，根据定义式容易发现 $f_d(z)=\mu(z)$。所以有：$ans=\sum_{i=1}^{n}\sum_{j=i}^{n}\mu(ij)$。

$$\mu(ij)=\mu(i)\mu(j)[\gcd(i,j)=1]$$

有相同因子的 μ 就是 0，所以只有两个数互质才可能有值，此时就可以利用积性函数的性质将 $\mu(ij)$ 拆开。接下来用一些基础的莫比乌斯反演就可以解决了。

$$\begin{aligned}
ans&=\sum_{i=1}^{n}\sum_{j=i}^{n}\mu(i)\mu(j)[\gcd(i,j)=1]\\
&=\sum_{i=1}^{n}\sum_{j=i}^{n}\mu(i)\mu(j)\sum_{t|\gcd(i,j)}\mu(t)\\
&=\sum_{t=1}^{n}\mu(t)\sum_{i=1}^{n}[t|i]\mu(i)\sum_{j=i}^{n}[t|j]\mu(j)
\end{aligned}$$

将每个数贡献到其约数位置上就能解决了。$[1,n]$ 的所有数的约数个数和是 $O(n\ln n)$。

用线性筛筛出 $\mu(i),i\in[1,n]$ 后，时间复杂度 $O(n\ln n)$。

2.5 subtask 5

subtask 4 的方法在 $d\geq 2$ 时无法进行拓展，必须采用新的方法。

容易发现若 $f_d(ij)$ 不为 0 时答案为 $g(ij)$，而 $g(ij)$ 是完全积性函数可以拆开，否则 $f_d(ij)^2$ 也为 0。故可以用以下式子来计算：

$$f_d(ij)=g(i)g(j)f_d(ij)^2$$

设 $h(x)=\max_{t\geq 1}\{[t^{d+1}|x]t\}$，即最大的 t 使得 t^{d+1} 是 x 的约数，于是有：

$$f_d(x)^2=[h(x)=1]=\sum_{t|h(x)}\mu(t)=\sum_{t^{d+1}|x}\mu(t)$$

考虑每次新加入一个数 v 时，和原来的数 $a_i,i\in[1,r]$ 在一起产生的贡献。

$$\begin{aligned}
ans&=\sum_{i=1}^{r}g(a_i)g(v)\sum_{t^{d+1}|a_iv}\mu(t)\\
&=g(v)\sum_{t}\mu(t)\sum_{i=1}^{r}g(a_i)[t^{d+1}|a_iv]\\
&=g(v)\sum_{t|v}\mu(t)\sum_{i=1}^{r}g(a_i)\left[\frac{t^{d+1}}{\gcd(t^{d+1},v)}|a_i\right]
\end{aligned}$$

关于最后一步 t 只用枚举到 v 的约数，因为若 t 不是 v 的约数又满足 $t^{d+1}|a_i v$，那么就一定存在一个质数 p 使得 $p^{d+1}|a_i$ 此时 $f_d(a_i)=0$。故忽略掉所有 $f_d(a_i)=0$ 的数后，就可以直接用最后一步的推法了。

依次加入每个数，在它的约数处贡献一下，就可以枚举约数快速计算答案。时间复杂度 $O(n\ln n)$。

如果你在 subtask 4 中采用 $\mu(ij)=\mu(i)\mu(j)\mu(ij)^2$ 的拆法，那么就可以用同样的做法快速拓展解决 subtask 5。具体推导方法和 subtask 5 一样，这里就不再赘述。

2.6 subtask 6

和 subtask 3 一样，先用 DFS 序将子树问题转为序列上的区间问题。

根据 subtask 5 的做法，可以快速加入一个数后维护答案，同理也可以用同样的方法快速去掉一个数后维护答案。

快速加入和删除数实现后，就可以套用普通莫队来求解这个区间问题。由于是子树转的区间，所以每个区间的左端点不同。给区间排序时，左端点的所在的块为第一关键字，右端点为第二关键字，这样就能保证每个点会加入删除 $O(\sqrt{n})$ 次。

时间复杂度 $O(n\ln n\sqrt{n})$。

2.7 subtask 7 标准解法

在 subtask 5 的基础上，可以用 dsu on tree。先进行重链剖分，每个点先继承其重儿子的答案，并继承记录下来的每个数对其约数的贡献数组，然后遍历其轻儿子所在的子树，一个一个将数加入后计算答案。

由于每个点到根的轻边个数是 $O(\log n)$ 的，所以每个点会被加入 $O(\log n)$ 次。时间复杂度 $O(n\ln n\log n)$。

还有另一种处理方法，和 subtask 3 一样，先用 DFS 序将子树问题转为序列上的区间问题。然后由于是子树转移过来的区间，所以两两区间要么是包含关系要么是不相交关系。

可以考虑分治，每次计算跨过 mid 的区间的答案。由于这些区间只可能是包含关系，所以只需要将区间排好序后，将维护的左右端点不断向外移动到各个区间所在位置，并计算答案即可。时间复杂度 $O(n\ln n\log n)$。

总时间复杂度 $O(n\ln n\log n)$，分治法常数没有 dsu on tree 优秀，所以最后 std 采用的是 dsu on tree 的做法。

3 总结

本题是一道莫比乌斯反演结合简单数据结构的题目。需要选手首先思考出函数的求法，然后经过简单的推演，最后用 dsu on tree 解决问题。

由于目前已有的题目中，用莫比乌斯反演推演式子后再结合非数学方面的知识解决的题目非常少。所以出题人做了这方面的一些尝试，将不同的知识板块结合到一起后命制出的本题。

总体上，本题思维难度适中，代码难度适中，可以较全面地考察选手的水平。

4 参考资料

无。