签到题

题目描述

给出一棵 n 个点 m 条边的仙人掌 G，点 i 有点权 a_i。（其中仙人掌定义为每条边至多出现在一个简单环中的**简单**无向连通图）。

q 次询问，每次询问给出 k，表示查询在仙人掌上选择一个非空子树（也就是仙人掌的一个无环连通子图）和子树中的一个点 r，使得子树中所有点到 r **在选出的子树上**的距离之和 sum 至少为 k 的情况下，子树上所有点的点权的 $\gcd$ 最大是多少。

输入格式

第一行 4 个数 C, n, m, q，其中 C 表示子任务编号（样例的 $C = 0$）。

接下来一行 n 个正整数 a_i。

接下来 m 行每行两个正整数 u, v 表示 u 和 v 由一条双向边链接。

接下来 q 行每行一个正整数 k。

以上字母未经特殊说明均与题目描述中含义一致。

输出格式

输出 q 行，每行一个数，依次表示对应询问的答案。特别地，如果不存在这样的选择子树方法，输出 `-1`。

样例输入

```
0 5 6 1
2 4 6 8 9
1 2
1 3
2 3
2 4
2 5
4 5
5
```

样例输出

```
2
```

更多样例见下发文件

数据范围

对于所有数据 $3 \le n \le 2 \times 10^5$，$n-1 \le m \le \min\{2n, 2 \times 10^5\}$，$1 \le a_i \le A \le 10^6$，$1 \le u, v \le n$，$1 \le q \le 2 \times 10^5$，$1 \le k \le 10^{10}$，保证输入是一棵仙人掌。

子任务编号	n	m	q	A	子任务依赖	分值
1	≤ 5		$=1$	$=20$		10
2	≤ 20		≤ 20	$=20$	1	10
3		$=n-1$				20
4		$=n$			3	20
5	≤ 100				2	20
6					4,5	20

表格中留空代表没有特殊限制。

题解做法

Sub 1

枚举仙人掌的所有非空子图和点 r，时间复杂度 $\mathrm{O}(\mathrm{poly}(n)2^{n+m})$。

Sub 2

使用状压 DP，设 f_S 表示集合 S 中的点被选入子树时距离之和 sum 最大是多少，枚举状态中每个点的所有出边进行转移。同时统计集合 S 中所有点点权的 $\gcd$ g_S 即可回答询问。

时间复杂度 $\mathrm{O}(m2^n)$。

问题的转化

由于询问的是最大 $\gcd$，可以考虑对于每个 w 求出满足 w 是答案的因数的最大限制 lim_w，则回答询问只需要查找最大的 w 使得 $lim_w \ge k$ 即可。

考虑找到所有点权是 w 倍数的点构成点集的导出子图 G_w，那么 lim_w 就等于在 G_w 中上选择子树的最大距离和 sum。因为根据定义，所有答案为 w 的倍数的选择子树方法一定是 G_w 的子树。

枚举所有的 w，问题转化为在每个 G_w 上求 sum 最大的子树选择方案。

结论：选择的子树点集一定是 G_w 上的某个极大联通块。

证明：对于点集 S 和 T（$S \subseteq T$，S 和 T 都是联通块），选择点集 S 的子树的最大 sum 一定不超过选择点集 T 的子树的最大 sum，因为只需把前一个的选边方案继承到 T 上就能得到一种不低于前一个 sum 的选择方案。

每个点 u 只会在 a_u 的所有因数 w 对应的 G_w 中出现，而每条边只会在两端点点权 $\gcd$ 的因数对应的图中出现，所以遍历所有 G_w 所花的时间复杂度是 $\mathrm{O}((n+m)\max_{1\le x\le A} d(x))$，其中 $d(x)$ 表示 x 的因数个数。当 $A = 10^6$ 时，$\max_{1\le x\le A} d(x) = 240$。

Sub 3

对于树的情况，只需要对每个联通块枚举所有可能的 r 并计算 sum。

算法一

考虑 r 沿边 (u,v) 从点 u 移向点 v 的过程，设断开 (u,v) 后 u,v 所在联通块大小分别为 su, sv，则移动后距离和 sum 的变化量为 $su - sv$。

先进行一次 dfs 求出 r 在点 1 处的答案和每个点的子树大小，再进行一次 dfs 模拟 r 的移动即可得到所有位置的答案。

算法二

算法一启示可以将贡献下放到边上，将所有点到 r 的距离之和转化为以 r 为根时所有边远离 r 一侧的子树大小之和。这样就可以使用换根 DP 统计贡献。

不妨取 1 号点作为根，设 f_u 表示 u 子树内所有点到 u 的距离之和，g_u 表示 u 子树外所有点到 u 的距离之和，tot 表示当前联通块的大小，siz_u 表示 u 子树内点的个数，fa_u 表示 u 的父亲，则有：

$$f_u = \sum_v f_v + siz_v$$

$$g_u = g_{fa_u} + tot - siz_u$$

则最终点 r 的 $\mathrm{sum}=f_r+g_r$。

Sub 4

对于基环树的情况，除了选择 r 以外，还要选择一条边断开。

考虑拓展 Sub 3 算法二的做法。对于不在环上的边，贡献的计算和 Sub 3 没有区别。对于在环上的部分，最终的贡献为每个子树的 siz 乘上断边后该子树到 r 所在子树在环上的距离，也就是说只取决于该子树在所断边的哪一侧。

结论：断掉的边一定是 r 所在子树旁边的两条边之一。

证明：考虑调整法，假设环长为 len，环上点从 r 所在子树对应点开始依次编号为 $0,1,\dots,len-1$，点 i 断开环上边后所在联通块大小为 siz_i，断开的边是 $(x,(x+1)\mod len)$。则贡献为

$$\sum_{i=1}^{x} siz_i i+\sum_{i=x+1}^{len-1} siz_i(len-i)$$

若 $0<x\le \frac{len}{2}$，则从 x 变为 $x-1$，贡献会变化 $-siz_i y+siz_i(len-x)=siz_i(len-2y)\ge 0$，调整不劣。

若 $\frac{len}{2}<x<len-1$，则从 x 变为 $x+1$，贡献变化 $-siz_{x+1}(len-x-1)+siz_{x+1}(x+1)=siz_{x+1}(2x-len)\ge 0$，调整不劣。

因此只需要记录 siz_i 和 $siz_i\times i$ 的前缀和计算贡献，取两者中较大值即可。

Sub 6

对于仙人掌的情况，拓展 Sub 4 中的做法，可以发现每个环上的贡献只与 r 在环上哪个点的子树内有关。因此建出圆方树，将 DP 定义中的子树改为圆方树上的子树即可。

时间复杂度 $\mathrm{O}((n+m)\max_{1\le x\le A} d(x))$，但常数较大，实现不够精细或处理部分不是严格线性也有一定可能通过。