

## Orac and LCM

题目链接<https://codeforc.es/contest/1350/problem/C>

### 题意

给定长为  $n$  ( $2 \leq n \leq 10^5$ ) 的正整数序列  $\{s_1, s_2, \dots, s_n\}$ , 求由任意两元素 LCM (最小公倍数) 组成新序列  $\{a_1, a_2, \dots, a_n\}$  ( $1 \leq a_i \leq 2 \cdot 10^5$ ) 的 GCD (最大公约数), 即  $\text{LCM}(\text{GCD}(s_i, s_j))$  ( $i \leq j$ )

### 题解

将答案  $ans$  质因数分解, 有  $ans = p_1^{k_1} p_2^{k_2} \dots p_m^{k_m}$

对于每个  $p_i$ , 一定在序列  $s$  中出现至少  $n - 1$  次, 每次出现由小到大可能为  $p_i^{c_1} p_i^{c_2} \dots p_i^{c_{ki}}$

$\text{GCD}(\text{LCM}(p_i^{c_1} p_i^{c_2} \dots p_i^{c_{ki}}))$  一定为第二小的  $c_2$

所以可以对序列  $s$  每个数的每个质因子统计次数, 将大于  $n-1$  次的因子幂次方排序, 取第二小累乘即可

注意用 vector 不要处处 long long 否则会 MLE

### AC代码

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;

const ll mod = 998244353;
const int maxn = 2e5 + 5;

vector<int> ep[maxn];

ll qpow(int k, ll b){
    ll res = 1;
    while(b){
        if(b & 1) res *= k;
        k *= k;
        b >>= 1;
    }
    return res;
}

int main(){
    int n;    scanf("%d", &n);
```

```
for(int i = 1; i <= n; i++){
    int t;    scanf("%d", &t);
    for(int i = 2; i * i <= t; ++i){
        ll cnt = 0;
        while(t % i == 0){
            t /= i;
            ++cnt;
        }
        ep[i].push_back(cnt);
    }
    if(t != 1) ep[t].push_back(1);
}

ll ans = 1;
for(int i = 1; i <= maxn; ++i){
    if(ep[i].size() >= n - 1){
        sort(ep[i].begin(), ep[i].end());
        if(ep[i].size() == n) ans *= qpow(i, ep[i][1]);
        else ans *= qpow(i, ep[i][0]);
    }
}
printf("%lld", ans);
}
```

From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: [https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:manespace:codeforc\\_round\\_641\\_div2\\_problem\\_c&rev=1589616807](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:manespace:codeforc_round_641_div2_problem_c&rev=1589616807)

Last update: 2020/05/16 16:13