有这样一个问题模型：

给定数a和b，问你是否存在k使得b^k%a==0

问题可以转化成，a的质因数种类，是否被b的质因数种类包含

a=40 b=100行

a=100 b=40行

a=3 b=30行

a=30 b=3 就不行，3的多少次方都不会整除30，30里包含了a里不存在的质因数5和2

数据组数T有很多组，T<=e5，且a,b<=1e18

## 暴力分解法:

分别对a和b分解质因数，再逐个比较a的质因数是否包含在b里面。

分解质因数不管你用sqrt(n)复杂度的试除法，还是快速分解法，都太慢

原因是数据太大，试除法每组测试数据需要1e9次，快速分解需要预处理质数，预处理质数要预处理1e9的次数，再用每组测试数据pow(1e18,1/4)复杂度分解，还是十分慢。所以暴力分解再比较的招法不行。

## 最大公约数法：

这是我当时想出来的，求bool isContain(ll a,ll b){

对a和b求最大公约数g,g一定包含了两者公共的质因数种类，但是a/g不一定是1，b/g也不一定是1，因为包含的公共质因数数量可能不同

然后呢，把g当做原本b,把a/g当做a,继续算，什么意思呢，因为我只关心a的质因数种类，看看能否被最大公约数g除尽

bool isContain(ll a,ll b){//a的质因数种类，是否被b的质因数种类包含

ll n=a,g=2;

for(n=a;g>1;){

g=gcd(n,b);

b=g;

n=n/g;

}

if(n==1){

return 1;

}

return 0;

}