题目需要找一个数至少能够被每对数中的1个数整除

也就是说，是每对数的一个因子

只需要将第一对数的质因子打出来，在判断剩下的数对中判断这些因子是不是他们的因子，

因为是大素数筛用的**Miller\_Rabin的筛法，其实普通暴力筛也能过**

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*

\* Miller\_Rabin �㷨������������

\* �ٶȿ죬�����ж�һ�� < 2^63 �����ǲ�������

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<bits/stdc++.h>

#include <cstdio>

#include <iostream>

#include <cstring>

#include <algorithm>

#include <cmath>

#include <ctime>

using namespace std;

const int S = 8;//����㷨�ж�������һ��8~10�ξ͹���

//����ret = (a\*b)%c a, b, c < 2^63

long long mult\_mod(long long a, long long b, long long c)

{

a %= c;

b %= c;

long long ret = 0;

long long tem = a;

while(b)

{

if(b & 1)

{

ret += tem;

if(ret > c) ret -= c;//ֱ��ȡģ���ܶ�

}

tem <<= 1;

if(tem > c) tem -= c;

b >>= 1;

}

return ret;

}

//���� ret = (a^n) % mod

long long pow\_mod(long long a, long long n, long long mod)

{

long long ret = 1;

long long tem = a % mod;

while(n)

{

if(n & 1) ret = mult\_mod(ret, tem, mod);

tem = mult\_mod(tem, tem, mod);

n >>= 1;

}

return ret;

}

// ͨ�� a^(n-1)=1(mod n)���ж�n�ǲ�������

// n-1 = x \* 2^t �м�ʹ�ö����ж�

// �Ǻ�������true����һ���Ǻ�������false

bool check(long long a, long long n, long long x, long long t)

{

long long ret = pow\_mod(a, x, n);//a^x % n

long long last = ret;

for(int i = 1; i <= t; ++i)//����t��(a^x % n)^2 % n

{

ret = mult\_mod(ret, ret, n);

if(ret == 1 && last != 1 && last != n - 1) return true;//����

last = ret;

}

if(ret != 1) return true;

return false;//��һ���Ǻ���

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// Miller\_Rabin�㷨

// ����������true,(������α����)

// ������������false

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

bool Miller\_Rabin(long long n)

{

if(n < 2) return false;

if(n == 2) return true;

if( (n&1) == 0) return false;//ż��

long long x = n - 1;

long long t = 0;

while( (x&1) == 0) //��n�ֽ�Ϊx\*2^t;

{

x >>= 1;

t++;

}

srand( time(NULL));

for(int i = 0; i < S; ++i)

{

long long a = rand()%(n-1) + 1;//���������a(�������䷶Χ��1 ~ n-1֮��)

if(check(a, n, x, t))//�Ǻ���

return false;

}

return true;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//

// pollard\_rho �㷨���������طֽ�

//

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int tol;//�������ĸ��������Ϊ0~tol-1

long long factor[100];//�����طֽ���(�շ���ʱ�������)

long long gcd(long long a, long long b)

{

long long t;

while(b)

{

t = a;

a = b;

b = t % b;

}

if(a >= 0) return a;

return -a;

}

//�ҳ�һ������

long long pollard\_rho(long long x, long long c)

{

long long i = 1, k = 2;

srand( time(NULL));

long long x0 = rand()%(x-1) + 1;//���������x0(�������䷶Χ��1 ~ x-1֮��)

long long y = x0;

while(1)

{

i++;

x0 = (mult\_mod(x0, x0, x) + c) % x;

long long d = gcd(y - x0, x);

if(d != 1 && d != x) return d;

if(y == x0) return x;

if(i == k)

{

y = x0;

k += k;

}

}

}

//��n���������ӷֽ⣬����factor�� k����Ϊ107���Ҽ���

void findfac(long long n, int k)

{

if(n == 1) return ;

if(Miller\_Rabin(n))//�������Ͱ���������Ӵ�����

{

factor[tol++] = n;

return ;

}

int c = k;

long long p = n;

while(p >= n)

p = pollard\_rho(p, c--);//ֵ�仯����ֹ������ѭ��k

findfac(p, k);

findfac(n/p, k);

}

long long x[151000],y[151000];

long long n;

bool check(long long tmp)

{

for(int i=0;i<n;i++)

if(x[i]%tmp!=0 && y[i]%tmp!=0)

return 0;

return 1;

}

int main()

{

int T;

scanf("%I64d",&n);

tol = 0;

for(int e=0;e<n;e++)

{

scanf("%I64d%I64d",&x[e],&y[e]);

}

findfac(x[0]\*y[0], 107);

sort(factor,factor+tol);

for(int i=0;i<tol;i++)

{

if(i==0 || factor[i]!=factor[i-1])

if(check(factor[i]))

{

printf("%I64d\n",factor[i]);

return 0;

}

}

printf("-1\n");

return 0;

}

法二：

x=gcd(lcm(a1,b1),lcm(a2,b2),lcm(a3,b3)

挑出来x因子中最小的就好

#include<bits/stdc++.h>

#include <cstdio>

#include <iostream>

#include <cstring>

#include <algorithm>

#include <cmath>

#include <ctime>

using namespace std;

typedef long long ll;

ll a[150010],b[150010];

int main()

{

int n;

scanf("%d",&n);

for(int i=0;i<n;i++)

{

scanf("%I64d%I64d",&a[i],&b[i]);

}

ll ans=a[0]\*b[0]/\_\_gcd(a[0],b[0]);

for(int i=1;i<n;i++)

{

ans=\_\_gcd(ans,a[i]\*b[i]/\_\_gcd(a[i],b[i]));

}

if(ans==1)

{

puts("-1");

return 0;

}

ll x;

for(int i=0;i<n;i++)

{

x=\_\_gcd(ans,a[i]);

if(x>1)

ans=x;

x=\_\_gcd(ans,b[i]);

if(x>1)

ans=x;

}

printf("%I64d\n",ans);

return 0;

}