# day06 题解

缪方岩

# 目录

# #A、又是约数个数

## 题目大意：

### 题目描述

给定正整数 $N$

记 $D_i$ 为正整数 $i$ 的约数个数

令

$$T_i = ((-1)^i \times (i + A)) \bmod P$$

> $\bmod$ *操作结果符号与第一个操作数符号相同,如* $-5 \bmod 7 = -5$ .

你需要求除

$$\sum_{i=1}^{N} (T_i \times D_i)$$

的结果

# #A、又是约数个数

思路：

很明显T[i]可以O(1)算出来，所以我们只要考虑D[i]即可

看一眼数据范围，可以联想到欧拉筛，那我们来研究一下如何用欧拉筛求出D[i]。

我们令i=$p_1{}^{c_1} * p_2{}^{c_2} * \cdots * p_{\mathrm{n}}{}^{c_n}$，

如果i为质数，很明显D[i]=2；

否则D[i]=($c_1$+1)*($c_2$+1)* ⋯ *($c_{\mathrm{n}}$+1)（注：这里的D[i]早已算好）。

接下来考虑的就是D[i*prime[j]]了；

如果prime[j] ∤ i，那么i*prime[j]= $prime_j{}^1 * p_1{}^{c_1} * p_2{}^{c_2} * \cdots * p_{\mathrm{n}}{}^{c_n}$，则D[i*prime[j]]=(1+1)*($c_1$+1)*($c_2$+1)* ⋯ *($c_{\mathrm{n}}$+1) =D[i]*2;

# #A、又是约数个数

思路：

如果prime[j] | i ，因为prime[i]是i的最小质因子（注1），所以 i*prime[i]= $p_1{}^{c_1+1} * p_2{}^{c_2} * \cdots * p_n{}^{c_n}$，D[i*prime[j]]=($c_1$+2)*($c_2$+1)* ⋯ *($c_n$+1)这个要怎么算出因数个数呢？

我们需要再开一个数组：cnt，cnt[i]表示i的最小质因子的系数（c1）

通过cnt[i]，我们可以算出($c_2$+1)* ⋯ *($c_n$+1)=D[i]÷(cnt[i]+1)，则 D[i*prime[j]]=D[i] ÷(cnt[i]+1)*(cnt[i+2])

注1：

证明：假设prime[i]不是i的最小质因子，即prime[i]≠p1，由于我们的prime数组是从小到大的，所以p1在prime[j]之前一定已经被遍历到了，又因为p1| i，所以循环在那时已经break了，不可能再遍历到prime[j]，假设不成立，证明完毕。

# #A、又是约数个数

思路:

这就已经结束了吗？NONONO，让我们来算一下空间复杂度:

| | |
|---|---|
| int prime[20000005] | 80MB |
| int D[20000005] | 80MB |
| int cnt[20000005] | 80MB |
| bool flag[20000005] | 19MB |

总计：约300MB

内存限制 128MB

所以，我们要开始“抠门”了！

1．2e7以内的质数没有2e7个，大约只有1.3e6个

2．cnt[i]最大只到25，所以可以只开char(1~128 绰绰有余)

3.1 flag的数据类型可以改成bitset(只有bool的32分之1)

3.2 flag也可以不用，只要判断D[i]==0即可；

再算一下空间，可以发现，已经不会MLE了！！！

这题才算结束。。。

# #A、又是约数个数

核心代码:

```
int prime[1300005];
int D[20000005];
char cnt[20000005];
bitset<20000005> f;
ll Euler(ll n) {
    ll size=0;
    for(ll i=2; i<=n; i++) {
        if(!f[i]) {
            prime[size++]=i;
            cnt[i]=1;
            D[i]=2;
        }
        for(ll j=0; j<size&&i*prime[j]<=n; j++) {
            f[i*prime[j]]=true;
            if(i%prime[j]) {
                D[i*prime[j]]=2*D[i];
                cnt[i*prime[j]]=1;
            } else {
                D[i*prime[j]]=D[i]/(cnt[i]+1)*(cnt[i]+2);
                cnt[i*prime[j]]=cnt[i]+1;
                break;
            }
        }
    }
    return size;
}
ll T(int i){
    return (i%2 ?-1 :1)*(i+a)%p;
}
```

```
Euler(n);
D[1]=1;
ll ans=0;
for(int i=1;i<=n;i++){
    ans+=T(i)*D[i];
}
```

# #B、裁剪正方形

## 题目大意：

### 题目描述

$Mas$ 拿到了一把剪刀,他需要将一个长为 $h$ 宽为 $w$ 的矩形进行裁剪

初始时称裁剪规则如下:

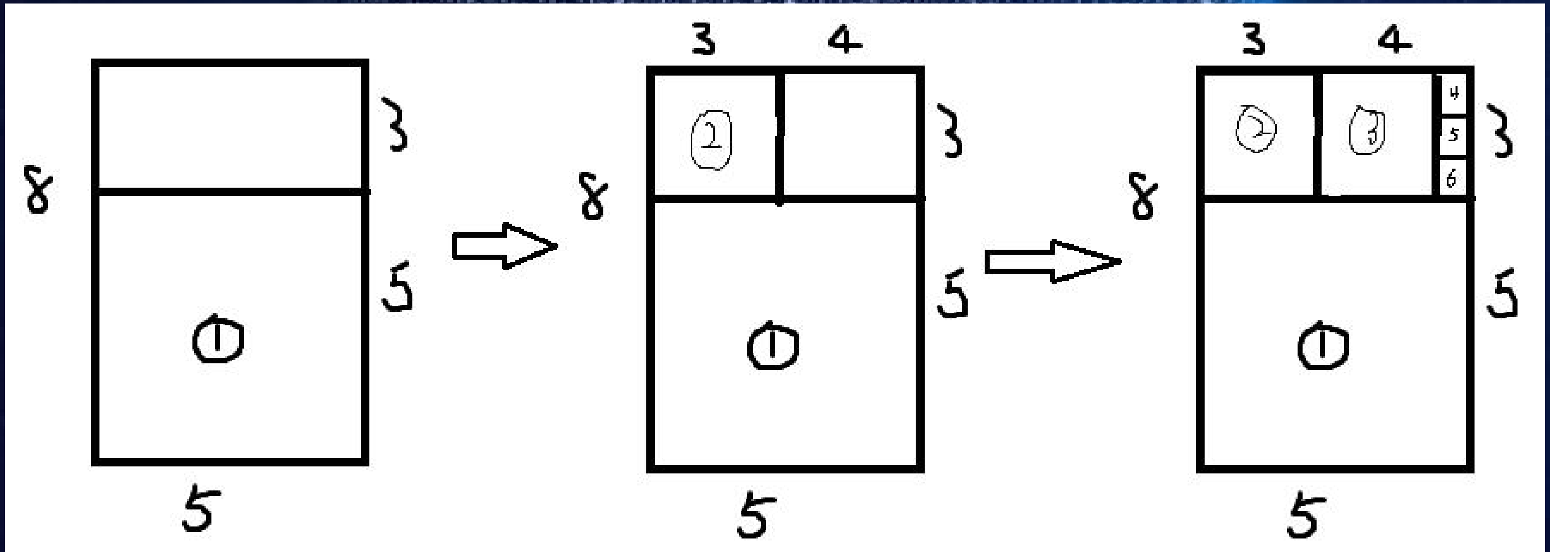
- 每次将矩形裁剪成一个正方形和一个矩形(剩余图形)
- 若剩余矩形为正方形,那么停止

为了尽快完成这个任务
请你计算 $Mas$ 需要进行多少次操作、最终能够裁剪出多少个正方形以及边长不同的正方形的个数

## #B、裁剪正方形

思路:

我们可以先模拟一下这个过程:




通过过程可以发现，这个过程是不是很像我们求gcd时用的辗转相除法!

我们可以在求gcd的过程中找到答案。

# #B、裁剪正方形

思路:

操作次数：每一层递归时的h/w累加后-1
不同边长的正方形数量：递归次数
最终正方形的个数：操作次数+1

核心代码:

```
ll gcd(ll a,ll b) {
    if (!b){
        return a;
    }
    ans1+=a/b;
    ans2++;
    return gcd(b,a%b);
}
```

```
if (h<w){
    swap(h,w);
}
gcd(h,w);
printf("%lld %lld %lld\n",ans1-1,ans2,ans1);
```

# #C、第K大公约数

**题目大意**：

T组数据，给定两个正整数A,B,输出第k大的公约数,如果没有，输出-1.

思路：

求出A,B的最大公约数n，只要有第k大的公约数，那它必然是n的因数（注）。那只要用O(√n)的时间复杂度求出k的所有因数，再从中找出第k大即可（可以用set维护）。

# #C、第K大公约数

注：证明一下这个结论：

令x=(a,b), y为a,b的任意公约数，
a=mx，b=nx，n,m互质
假设y不是x的因数，则y必然含有x所没有的因数t
因为a mod y=0，b mod y=0
所以a mod t=0， b mod t=0
又因为x mod t≠0，所以n mod t=0，m mod t=0
那t为n,m的公约数
与 n,m互质 矛盾，假设不成立，所以y必然是x的因数

# #D、加值GCD

## 题目大意:

给你两个正整数数组 $a_1, a_2, \cdots, a_n$ 和 $b_1, b_2, \cdots, b_m$

请你求出 $a_1 + b_i, a_2 + b_i, \cdots, a_n + b_i$ 的最大公约数

## 思路:

我们都知道:

gcd(a,b)=gcd(a,b-a)=gcd(a,a-b)=gcd(b,a-b)=gcd(b,b-a)

那gcd(a,b,c)呢？

gcd(a,b,c)=gcd(a,b-a,b,c-b)=gcd(a,b-a,c-b)

再推广到n个数字，就是:

gcd(a[1],a[2],…,a[n])=gcd(a[1],a[2]-a[1],a[3]-a[2],…,a[n]-a[n-1])

# #D、加值GCD

思路:

将题目要求的内容套入，可以发现:
gcd(a[1]-b[i],a[2]-b[i],…a[n]-b[i])
= gcd(a[1]-b[i],a[2]-b[i]-a[1]-b[i],…a[n]-b[i]-a[n-1]-b[i])　　　b[i]抵消
=gcd (a[1]-b[i],a[2-a[1],…a[n]-a[n-1])!!!!
所以，我们只要在输入a[i]时把gcd(a[2-a[1],…a[n]-a[n-1])算出来。　(x)
再在输入b[i]时输出gcd(a[1]-b[i],x)即可

# #D、加值GCD

核心代码:

```
for(ll i=1;i<=n;i++){
    a[i]=read();
    if(i>1){
        ll t=abs(a[i]-a[i-1]);
        if(!x){
            x=t;
        }
        else{
            x=__gcd(x,t);
        }
    }
}
for(ll i=1;i<=m;i++){
    ll t=read();
    t+=a[1];
    ll ans=__gcd(x,t);
    cout<<ans<<endl;
}
```