



链滴

## 从零开始的算法题生活 (三)

作者: [yiranblade](#)

原文链接: <https://ld246.com/article/1491450606728>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 引言

练习的有点频繁了，无奈近来心情太乱，刷刷算法题有益身体健康^\_^,本次的题是hihoCoder144周习题

## 描述

小Hi最近在追求一名学数学的女生小Z。小Z其实是想拒绝他的，但是找不到好的说辞，于是提出了这的要求：对于给定的两个正整数N和M，小Hi随机选取一个N的约数N'，小Z随机选取一个M的约数M'，如果N'和M'相等，她就答应小Hi。

小Z让小Hi去编写这个随机程序，到时候她review过没有问题了就可以抽签了。但是小Hi写着写着，越来越觉得机会渺茫。那么问题来了，小Hi能够追到小Z的几率是多少呢？

## 输入

每个输入文件仅包含单组测试数据。

每组测试数据的第一行为两个正整数N和M，意义如前文所述。

对于40%的数据，满足 $1 \leq N, M \leq 106$

对于100%的数据，满足 $1 \leq N, M \leq 1012$

## 输出

对于每组测试数据，输出两个互质的正整数A和B（以A分之B表示小Hi能够追到小Z的几率）。

样例输入

3 2

样例输出

4 1

##解析

提取关键信息后，本题可抽象为给出两个数，求出一个数约数和另一个约数的所有组合，组合中约数等的组合占所有组合的比例。

很容易得到以下解题思路：

- 1、先求出两数约数不同组合个数记为sum
- 2、找出相等的组合个数记为count
- 3、 $\text{count}/\text{sum}$ ,化到最简

代码如下:(gcc6.3.1)

```
#include<stdio.h>
#include<stdlib.h>
typedef long long LL;
#define max(x,y) x>y?x:y
```

```

#define min(x,y) x>y?y:x
LL getCount(LL N,LL * array);
int main(void){
    LL N,M,N1,M1;
    LL count=0;
    LL sum=0;
    LL sum1=0;
    LL temp=0;
    LL i=0;
    LL j=0;
    LL arrayn[100000]={0};
    LL arraym[100000]={0};
    scanf("%lld%lld",&N,&M);
    N1=getCount(N,arrayn);
    M1=getCount(M,arraym);
    sum=N1*M1;
    for(i=1;i<=N1;i++){
        for(j=1;j<=M1;j++){
            if(arrayn[i]==arraym[j]){
                count++;
            }
        }
    }
    temp=max(sum,count);
    while(temp>=1){
        if(sum%temp==0&&count%temp==0){
            printf("%lld %lld",sum/temp,count/temp);
            break;
        }
        else temp--;
    }
    return 0;
}
LL getCount(LL N,LL *array){
    LL n=N;
    LL count=0;
    LL i;
    LL j;
    if (n%2==0)
    {
        for (i = 1,j=1; i <= n / 2; i++,j++){
            if (n % i == 0){
                array[j]=i;
                count++;
            }
        }
    }
    array[n]=1;
    count++;
}
if (n%2==1){
    for (i=1,j=1;i<=(n-1)/2;i++,j++){
        if (n % i == 0){

```

```

        array[j]=i;
        count++;
    }
}
array[n]=1;
count++;
}
return count;
}

```

然而这段代码的运行结果没有通过所有的测试用例，即没有AC，自己阅读原题后发现对于100%的数，满足 $1 \leq N, M \leq 10^{12}$ ，这就很尴尬了，数组大小申请空间肯定不够。又不想去自己实现确保数组请到那么大的空间，所以想到利用C++中的map去搞，，终于AC，代码如下：

```

#include<cstdio>
typedef long long LL;
#include<map>
using namespace std;
map<LL,int> arrays;
LL sum=0;
int flag=0;
#define max(x,y) x>y?x:y
#define min(x,y) x>y?y:x
LL getCount(LL N);
int main(void){
    LL N,M,N1,M1;
    LL count=0;
    LL temp=0;
    LL minvalue=0;
    LL i=0;
    scanf("%lld%lld",&N,&M);
    N1=getCount(N);
    M1=getCount(M);
    count=N1*M1;
    minvalue=min(N,M);
    temp=max(sum,count);
    while(temp>=1){
        if(sum%temp==0&&count%temp==0){
            printf("%lld %lld\n",count/temp,sum/temp);
            break;
        }
        else temp--;
    }
    return 0;
}
LL getCount(LL N){
    LL n=N;
    LL count=0;
    LL i;
    for (i = 1; i*i <= n ; i++){
        if (n % i == 0){
            if(flag==0){

```

```
        arrays[i]=1;
        count++;
    if(i!=n/i){
        arrays[n/i]=1;
        count++;
    }
    }else{
        count++;
        if(arrays[i])
            sum++;
        if(i!=n/i){
            count++;
            if(arrays[n/i])
                sum++;
        }
    }
}
}
}
flag=1;
return count;
}
//关于C++ map的使用遍历有兴趣可以看下实现源码
```