long-long-ago null

目录视图

个人资料

long-long-ago

关注 发私信

访问： 66307次
积分： 3467
等级： BLOG 5
排名： 第8645名

原创： 270篇 转载： 48篇
译文： 0篇 评论： 7条

模线性方程组

2016-05-07 22:19 715人阅读 评论

分类： 数论（59） algorithm（56）

先说一个故事

说秦末，刘邦的将军韩信带领1500名士兵经历了一场战斗，战死四百余人。韩信为了清点人数让士兵站成三人一排，多出来两人四人；站成七人一排，多出来六人。韩信立刻就知道了剩余人数为1049人。

这就是著名的韩信点兵的故事，化成数学模型就是：

韩信是为了计算的是士兵的人数，那么我们设这个人数为x。三人成排，五人成排，七人成排，即x mod 3, x mod 5, x mod 7。

组方程：

```
x mod 3 = 2

x mod 5 = 4

x mod 7 = 6
```

将这个方程组推广到一般形式：**给定了n组除数m[i]和余数r[i]，通过这n组(m[i],r[i])求解一个x，使得x mod m[i] = r[i]**

这就是**模线性方程组**

一开始就直接求解多个方程不是太容易，我们从n=2开始递推：

已知：

```
x mod m[1] = r[1]
x mod m[2] = r[2]
```

根据这两个式子，我们存在两个整数k[1],k[2]：

```
x = m[1] * k[1] + r[1]
x = m[2] * k[2] + r[2]
```

由于两个值相等，因此我们有：

```
    m[1] * k[1] + r[1] = m[2] * k[2] + r[2]
=>  m[1] * k[1] - m[2] * k[2] = r[2] - r[1]
```

由于m[1],m[2],r[1],r[2]都是常数，若令A=m[1],B=m[2],C=r[2]-r[1],x=k[1],y=k[2]，则上式变为：Ax + By = C。

这就转换成扩展的欧几里得算法

可以先通过gcd(m[1], m[2])能否整除r[2]-r[1]来判定是否存在解。

假设存在解，则我们通过扩展欧几里德求解出k[1],k[2]。

再把k[1]代入x = m[1] * k[1] + r[1]，就可以求解出x。

2016年10月 (23)
2016年09月 (2)
2016年08月 (6)
2016年07月 (4)

展开

阅读排行

介绍几款Web服务器性能压力... (2976)
判断素数的几种方法的总结 (2595)
MySQL 5.6.24 for Windows ... (1894)
C++ cout格式化输出（转） (1368)
Java 正则表达式 (1159)
工具使用，PS万能对齐大法 (1144)
2015郑州大学ACM暑期集训—.. (914)
调色技巧，超实用的可选颜色... (876)
Codeforces Round #340 (D... (820)
Wunder Fund Round 2016 (... (786)

评论排行

判断素数的几种方法的总结 (3)
2015ZZUACM暑期集训 (1)
程序员装逼指南（转） (1)
Wunder Fund Round 2016 (... (1)
HNACM(八)D-引水工程 (1)
poj 3984迷宫问题 (0)
Codeforces548C:Mike and ... (0)
POJ3104 Drying (0)
DBCP和Druid数据库连接池使... (0)
poj 1089 Intervals (0)

推荐文章

* 5月书讯：流畅的Python，终于等到你！
* 【新收录】CSDN日报 —— Kotlin 专场
* Android中带你开发一款自动爆破签名校验工具kstools
* Android图片加载框架最全解析——深入探究Glide的缓存机制
* Android 热修复 Tinker Gradle Plugin解析
* Unity Shader-死亡溶解效果

最新评论

判断素数的几种方法的总结
帝狱大大 ：那个miller_rabin有问题，6481 90992254089，应该是素数。。。一直过不掉。

HNACM(八)D-引水工程
帝狱大大 ：这个题我的思路是把自建费用当做结点0对应的，p，p...然后整个图找最小生成树。。白书配套的训练指南...

判断素数的几种方法的总结
long-long-ago ：@qq_30368701:已经改过来了，不过也有一个小问题当N非常大的时候，假如N是1e6+1，当i...

判断素数的几种方法的总结
帝狱大大 ：Eratosthenes筛法第10行，j从i * i开始，小于i * i 的都已经在前面筛过了

2015ZZUACM暑期集训
帝狱大大 ：这是耿学姐讲的，我都看出来了。。=.=

Wunder Fund Round 2016 (Div. 1 + Di...
Ginray ：坐等D、E、F、G ：）

程序员装逼指南（转）
wzw_ice ：程序猿 不用这些的

同时我们将这个x作为特解，可以扩展出一个解系：

```
X = x + k*lcm(m[1], m[2]) k为整数
```

lcm(a,b)表示a和b的最小公倍数。其求解公式为lcm(a,b)=a*b/gcd(a,b)。

将其改变形式为：

```
X mod lcm(m[1], m[2]) = x。
```

令M = lcm(m[1], m[2]), R = x，则有新的模方程X mod M = R。

此时，可以发现我们将x mod m[1] = r[1]，x mod m[2] = r[2]合并为了一个式子X mod lcm(m[1], m[2]) = x。满足后者的X一

每两个式子都可以通过该方法化简为一个式子。那么我们只要重复进行这个操作，就可以将n个方程组化简为一个方程，并且求

最后代码就是

```cpp
#include <iostream>
#include <cstdio>
#define N 1010
using namespace std;
long long m[N], r[N];
int n;
long long x, y;
long long gcd(long long a, long long b){
    if (!b) return a;
    return gcd(b, a%b);
}
pair<long long, long long > extend_gcd(long long a, long long b){
    pair<long long, long long> tmp;
    if (a%b == 0){
        return pair<long long , long long>(0, 1);
    }
    tmp = extend_gcd(b, a%b);
    long long tmp_x = tmp.first, tmp_y = tmp.second;
    x = tmp_y;
    y = tmp_x-(a/b)*tmp_y;
    return pair<long long , long long>(x, y);
}
long long solve(){
    long long M = m[0], R = r[0];
    for (int i = 1; i < n; i++){
        long long d = gcd(M, m[i]);
        long long c = r[i] - R;
        if (c%d != 0){
            return -1;  // 无解的情况
        }
        pair<long long, long long> t = extend_gcd(M/d, m[i]/d); // 扩展欧几里德计算
        t.first = (c/d*t.first)%(m[i]/d);   // 扩展解系
        R = R+t.first*M;
        M = M/d*m[i];   // 求解lcm(M, m[i])
        R %= M;     // 求解合并后的新R，同时让R最小
    }
    if (R < 0){
         R += M;
    }
    return R;
}
int main()
{
#ifndef ONLINE_JUDGE
    freopen("1.txt", "r", stdin);
#endif
    int i, j, k;
    scanf("%d", &n);
    for (i = 0; i < n; i++){
        scanf("%d%d", &m[i], &r[i]);
    }
    cout << solve();
    return 0;
}
```

- 上一篇 数论知识总结
- 下一篇 poj-1149-PIGS(最大流)

相关文章推荐

- 【学习笔记】Get Started with MATLAB-Chapter03
- PKU_算法_分类
- pku算法
- POJ 超详细分类
- poj题目分类
- poj 超详细分类
- PKU_算法_分类
- POJ前面的题目算法思路【转】
- ACM代码库
- 20世纪最好的十大算法、算法笔记（2008-11-15 22

参考知识库

算法与数据结构知识库

16385 关注 | 2320 收录

猜你在找

《C语言/C++学习指南》加密解密…
C语言系列之 字符串相关算法
C语言系列之 数组与算法实战
C语言系列之 递归算法示例与 W…
C语言系列之 字符串压缩算法与结…
C语言系列之 快速排序与全排列算…
模板匹配的字符识别(OCR)算法原理
ArcGIS for javascript 项目实战…
ArcGIS for JavaScript
C# For Unity系列之基础篇

查看评论

暂无评论

您还没有登录,请[登录]或[注册]

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack V
IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery BI HTML5 Spring Ap
API HTML SDK IIS Fedora XML LBS Unity Splashtop UML components Windows Mob
QEMU KDE Cassandra CloudStack FTC coremail OPhone CouchBase 云计算 iOS6 Rackspa
SpringSide Maemo Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HB
Angular Cloud Foundry Redis Scala Django Bootstrap