

个人资料



ACdreamers

访问：2199397次

积分：23616

等级：BLOG > 7

排名：第301名

原创：478篇 转载：42篇

译文：0篇 评论：488条

文章搜索

文章分类

数论 (69)

图论 (30)

搜索 (14)

字符串 (22)

基础数学 (76)

计算几何 (40)

组合数学 (27)

动态规划 (28)

数据结构 (61)

文学类 (39)

C/C++ (29)

HTML5 (8)

Python (17)

Java编程 (15)

人工智能 (24)

技术拓展 (9)

数学之美 (11)

文章存档

2015年10月 (1)

2015年06月 (4)

2015年05月 (3)

2015年04月 (4)

异步赠书：10月Python畅销书升级    【线路图】人工智能到底学什么？！    程序员9月书讯    每周荐书（京东篇）：618取胜之道、质量保障、技术解密）

中国剩余定理

2012-10-08 21:20    18900人阅读    评论(1)

分类：数论 (68)

版权声明：本文为博主原创文章，未经博主允许不得转载。

### 中国剩余定理（CRT）的表述如下

设正整数 $m_1, m_2, \dots, m_k$ 两两互素，则同余方程组

$$\begin{aligned}x &\equiv a_1(mod\ m_1) \\x &\equiv a_2(mod\ m_2) \\x &\equiv a_3(mod\ m_3) \\&\vdots \\x &\equiv a_k(mod\ m_k)\end{aligned}$$

有整数解。并且在模 $M = m_1 \cdot m_2 \cdot \dots \cdot m_k$ 下的解是唯一的，解为

$$x \equiv (a_1M_1M_1^{-1} + a_2M_2M_2^{-1} + \dots + a_kM_kM_k^{-1})mod\ M$$

其中 $M_i = M/m_i$ ，而 $M_i^{-1}$ 为 $M_i$ 模 $m_i$ 的逆元。

代码：

```
[cpp]
01. int CRT(int a[],int m[],int n)
02. {
03.     int M = 1;
04.     int ans = 0;
05.     for(int i=1; i<=n; i++)
06.         M *= m[i];
07.     for(int i=1; i<=n; i++)
08.     {
09.         int x, y;
10.         int Mi = M / m[i];
11.         extend_Euclid(Mi, m[i], x, y);
12.         ans = (ans + Mi * x * a[i]) % M;
13.     }
14.     if(ans < 0) ans += M;
15.     return ans;
16. }
```

## 阅读排行

|            |         |
|------------|---------|
| BP神经网络     | (85576) |
| 模拟退火算法     | (59194) |
| 决策树之ID3算法  | (42663) |
| 莫比乌斯反演     | (37738) |
| 协同过滤算法     | (37679) |
| 决策树之CART算法 | (35164) |
| softmax回归  | (34118) |
| 逆元详解       | (33285) |
| 相对熵 (KL散度) | (33284) |
| 石子合并问题     | (28234) |

## 评论排行

|            |      |
|------------|------|
| BP神经网络     | (33) |
| 深度理解链式前向星  | (24) |
| 莫比乌斯反演     | (17) |
| 网络刷博器      | (15) |
| 决策树之ID3算法  | (14) |
| softmax回归  | (13) |
| 决策树之CART算法 | (12) |
| 协同过滤算法     | (12) |
| 逆元详解       | (12) |
| BFGS算法     | (11) |

## 推荐文章

- \* CSDN邀请您来GitChat赚钱啦！
- \* 行为驱动开发 (BDD) 你准备好了吗？
- \* 如何更加安全、高效地利用开源项目？
- \* 程序员业余时间修炼指南
- \* DevOps 在公司项目中的实践落地
- \* Jenkins + Django 完整实战，细化到每一步操作

## 最新评论

- HDU4372(第一类斯特林数)  
milesgu: @Flynn\_curry:代码确实有问题,需要特判x+y>n+1的时候输出0
- 逆元详解  
Ifb637: 能不能转载一下,方便学习
- 深度理解链式前向星  
伪学渣: 意外发现一个好博客
- K-D树  
yxlshk: C++那一版的建树时,没有按照方差大小选举用来比较的维度? 目测是根据深度依次选取维度。
- 中国剩余定理  
海边拾贝的言: poj2891 可以用欧几里得扩展直接求吧

题目：<http://poj.org/problem?id=1006>

题意：人自出生起就有体力，情感和智力三个生理周期，分别为23，28和33天。一个周期内有一天为峰值，在这一天，人在对应的方面（体力，情感或智力）表现最好。通常这三个周期的峰值不会是同一天。现在给出三个日期，分别对应于体力，情感，智力出现峰值的日期。然后再给出一个起始日期，要求从这一天开始，算出最少再过多少天后三个峰值同时出现。

代码：

```
[cpp]
01. #include <iostream>
02. #include <string.h>
03. #include <stdio.h>
04.
05. using namespace std;
06.
07. int a[4], m[4];
08.
09. void extend_Euclid(int a, int b, int &x, int &y)
10. {
11.     if(b == 0)
12.     {
13.         x = 1;
14.         y = 0;
15.         return;
16.     }
17.     extend_Euclid(b, a % b, x, y);
18.     int tmp = x;
19.     x = y;
20.     y = tmp - (a / b) * y;
21. }
22.
23. int CRT(int a[],int m[],int n)
24. {
25.     int M = 1;
26.     int ans = 0;
27.     for(int i=1; i<=n; i++)
28.         M *= m[i];
29.     for(int i=1; i<=n; i++)
30.     {
31.         int x, y;
32.         int Mi = M / m[i];
33.         extend_Euclid(Mi, m[i], x, y);
34.         ans = (ans + Mi * x * a[i]) % M;
35.     }
36.     if(ans < 0) ans += M;
37.     return ans;
38. }
39.
40. int main()
41. {
42.     int p, e, i, d, t = 1;
43.     while(cin>>p>>e>>i>>d)
44.     {
45.         if(p == -1 && e == -1 && i == -1 && d == -1)
46.             break;
47.         a[1] = p;
48.         a[2] = e;
49.         a[3] = i;
50.         m[1] = 23;
51.         m[2] = 28;
52.         m[3] = 33;
```

0x5f3759d的数学原理

lindexi\_gd: 厉害

决策树之ID3算法

Timmy\_Y: 学习了

ISODATA算法

维克多、宇哥: @wzr22:运行可以按ctrl+f5,而不是只按f5

二次同余方程的解

墙角的花: @u010510549:x一定是整数,所以 $(a+\sqrt{w})^{((p-1)/2)}$ 也一定是整数,只...

数学公式及定理

snbsczw: 大神你好 第一个公式把n=1,k=2,带入怎么算的是0

```
53.         int ans = CRT(a, m, 3);
54.         if(ans <= d)
55.             ans += 21252;
56.         cout<<"Case "<<t++<<": the next triple peak occurs in "<<ans - d<<" days."<<endl;
57.     }
58.     return 0;
59. }
```

普通的中国剩余定理要求所有的  $m_i$  互素,那么如果不互素呢,怎么求解同余方程组?

这种情况就采用两两合并的思想,假设要合并如下两个方程

$$x = a_1 + m_1 x_1$$

$$x = a_2 + m_2 x_2$$

那么得到

$$a_1 + m_1 x_1 = a_2 + m_2 x_2 \Rightarrow m_1 x_1 + m_2 x_2 = a_2 - a_1$$

在利用扩展欧几里得算法解出  $x_1$  的最小正整数解,再带入

$$x = a_1 + m_1 x_1$$

得到  $x$  后合并为一个方程的结果为

$$y \equiv x \pmod{\text{lcm}(m_1, m_2)}$$

这样一直合并下去,最终可以求得同余方程组的解。

题目: <http://poj.org/problem?id=2891>

代码:

```
[cpp]
01. #include <iostream>
02. #include <string.h>
03. #include <stdio.h>
04.
05. using namespace std;
06. typedef long long LL;
07. const int N = 1005;
08.
09. LL a[N], m[N];
10.
11. LL gcd(LL a, LL b)
12. {
13.     return b? gcd(b, a % b) : a;
14. }
15.
16. void extend_Euclid(LL a, LL b, LL &x, LL &y)
17. {
18.     if(b == 0)
19.     {
20.         x = 1;
21.         y = 0;
22.         return;
23.     }
24.     extend_Euclid(b, a % b, x, y);
```

```

25.     LL tmp = x;
26.     x = y;
27.     y = tmp - (a / b) * y;
28. }
29.
30. LL Inv(LL a, LL b)
31. {
32.     LL d = gcd(a, b);
33.     if(d != 1) return -1;
34.     LL x, y;
35.     extend_Euclid(a, b, x, y);
36.     return (x % b + b) % b;
37. }
38.
39. bool merge(LL a1, LL m1, LL a2, LL m2, LL &a3, LL &m3)
40. {
41.     LL d = gcd(m1, m2);
42.     LL c = a2 - a1;
43.     if(c % d) return false;
44.     c = (c % m2 + m2) % m2;
45.     m1 /= d;
46.     m2 /= d;
47.     c /= d;
48.     c *= Inv(m1, m2);
49.     c %= m2;
50.     c *= m1 * d;
51.     c += a1;
52.     m3 = m1 * m2 * d;
53.     a3 = (c % m3 + m3) % m3;
54.     return true;
55. }
56.
57. LL CRT(LL a[], LL m[], int n)
58. {
59.     LL a1 = a[1];
60.     LL m1 = m[1];
61.     for(int i=2; i<=n; i++)
62.     {
63.         LL a2 = a[i];
64.         LL m2 = m[i];
65.         LL m3, a3;
66.         if(!merge(a1, m1, a2, m2, a3, m3))
67.             return -1;
68.         a1 = a3;
69.         m1 = m3;
70.     }
71.     return (a1 % m1 + m1) % m1;
72. }
73.
74. int main()
75. {
76.     int n;
77.     while(scanf("%d",&n)!=EOF)
78.     {
79.         for(int i=1; i<=n; i++)
80.             scanf("%I64d%I64d",&m[i], &a[i]);
81.         LL ans = CRT(a, m, n);
82.         printf("%I64d\n",ans);
83.     }
84.     return 0;
85. }

```

**题目：**<http://acm.hdu.edu.cn/showproblem.php?pid=1573>

**分析：**这个题由于数据范围小，那么直接可以通过枚举在这  $m$  个数的最小公倍数范围内的所有数，找到最小的正整

数解，然后后面的所有解都可以通过这个得到。

## 代码：

```
[cpp]

01. #include <iostream>
02. #include <string.h>
03. #include <stdio.h>
04.
05. using namespace std;
06. const int N = 25;
07.
08. int a[N], b[N];
09.
10. int gcd(int a, int b)
11. {
12.     return b ? gcd(b, a % b) : a;
13. }
14.
15. int main()
16. {
17.     int T;
18.     cin>>T;
19.     while(T-->0)
20.     {
21.         int n, m;
22.         cin>>n>>m;
23.         for(int i=0; i<m; i++)
24.             cin>>a[i];
25.         for(int i=0; i<m; i++)
26.             cin>>b[i];
27.         int lcm = 1;
28.         for(int i=0; i<m; i++)
29.             lcm = lcm / gcd(lcm, a[i]) * a[i];
30.         bool f = 1;
31.         for(int i=1; i<=lcm&&i<=n; i++)
32.         {
33.             f = 1;
34.             for(int j=0; j<m; j++)
35.             {
36.                 if(i % a[j] != b[j])
37.                     f = 0;
38.             }
39.             if(f)
40.             {
41.                 printf("%d\n", (n - i) / lcm + 1);
42.                 break;
43.             }
44.         }
45.         if(f == 0)
46.             printf("0\n");
47.     }
48.     return 0;
49. }
```

顶

19

踩

1

[上一篇](#) 二维树状数组

[下一篇](#) PollardRho大整数分解

相关文章推荐

- OJ C++读取数据 cin关闭同步
- 用户画像系统应用与技术解析--汪剑
- c++中关于cin.tie()以及sync\_with\_stdio同步
- 2017 求职面试集训营之VIP服务版--刘道宽
- POJ-DP题目列表【开启疯狗模式】
- 如何优雅地编写Java
- 中国剩余定理（详解）
- MySQL特性详解
- SG函数模板
- React全家桶之Web基础应用
- 对于SG函数的理解
- OpenStack从入门到放弃
- 莫比乌斯反演
- hdu 6004 Periodical Cicadas [2016 CCPC-Final ...
- 只有20%的iOS程序员能看懂：详解intrinsic
- 中国剩余定理——另一种证明

查看评论

4楼 [海边拾贝的言](#) 2017-10-07 21:00发表



poj2891 可以用欧几里得扩展直接求吧

3楼 [OrdinaryCrazy](#) 2017-09-10 20:31发表



博主，能做OpenJudge百炼2793吗？也是不互质的情况

2楼 [sinat\\_35951073](#) 2016-08-24 10:21发表



应该是 $a1+m1*x1=a2+m2*x2 \rightarrow m1*x1-m2*x2=a2-a1$   
这里好像打错了= =

Re: [just\\_sort](#) 2017-01-13 16:43发表



回复sinat\_35951073：这个是没错的，m1和m2可以取任意整数，负无穷到正无穷反过来是一样的。

1楼 [n-1](#) 2014-10-08 21:48发表



博主。merge（）函数里面能解释一下吗

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