{% note info %} **摘要** Title: 204. 表达整数的奇怪方式 Tag: 中国剩余定理、二元一次不定方程通解、取模问题、最小正整数解 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by:**NEFU AB-IN**

Link

@TOC

204. 表达整数的奇怪方式

- 题意

> 给定 2n 个整数 a1,a2,…,an 和 m1,m2,…,mn， 求一个最小的非负整数 x，满足 ∀i∈[1,n],x≡ai(mod mi)。

- 思路

  {% note info %} **注意** 以下为了方便，**m和a反过来了** {% endnote %}

  前置知识: **二元一次不定方程通解**

  初等数论--同余方程--二元一次不定方程的通解形式

  博主是初学初等数论（整除+同余+原根），本意是想整理一些较难理解的定理、算法，加深记忆也方便日后查找；如果有错，欢迎指正。
  我整理成一个系列：初等数论，方便检索。

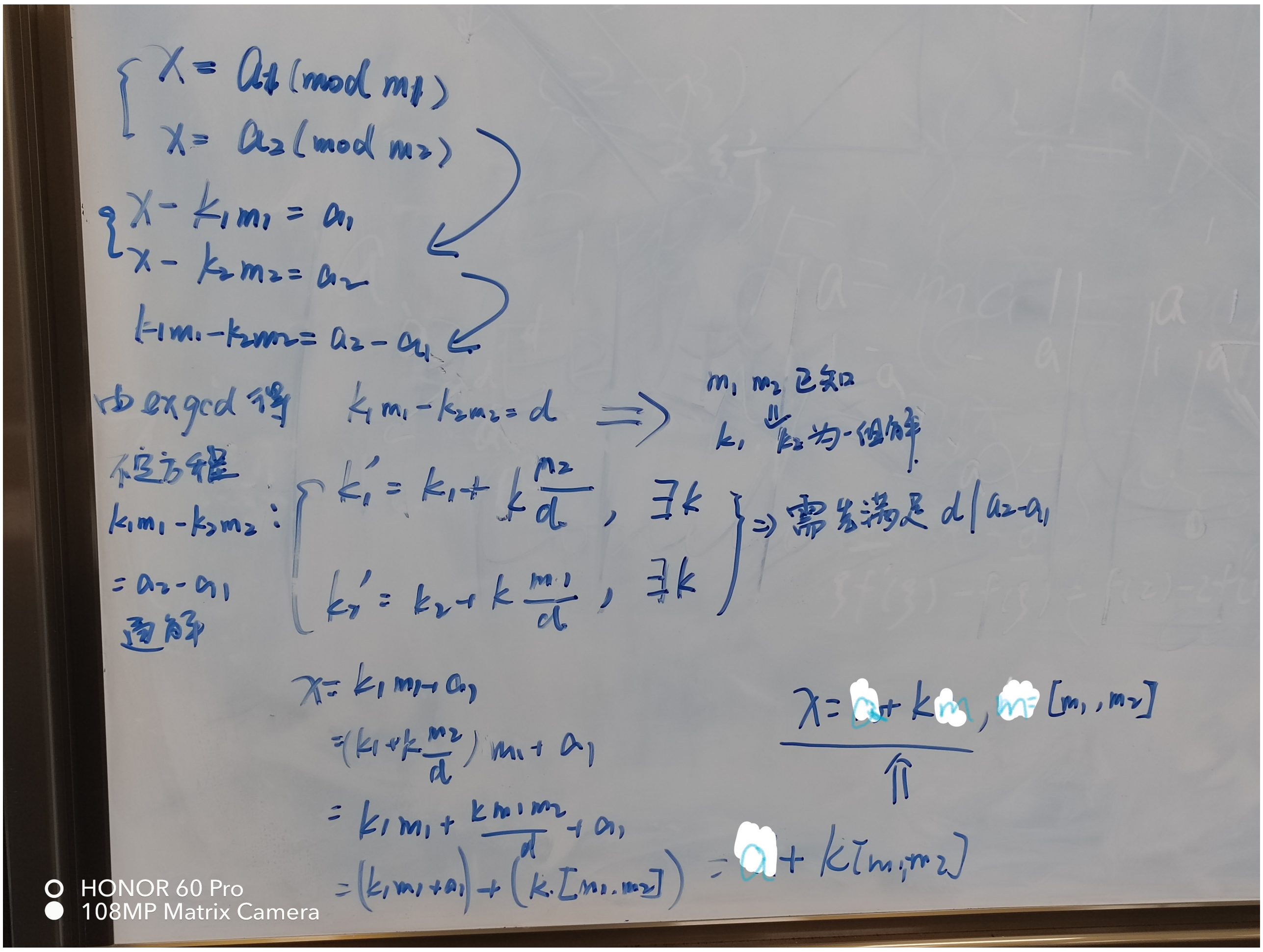
  - 不定方程：变量个数>方程个数

  若二元一次不定方程$ax+by=n$有解，x_0, y_0为它的一组整数解，则通解为

$$\begin{cases} x = & x_0 + \dfrac{b}{(a,b)} \cdot t \\ y = & y_0 - \dfrac{a}{(a,b)} \cdot t \end{cases} t \in Z$$

  可以发现这个题的m不都互质，所以不能直接采用中国剩余定理，那么就可以推导**中国剩余定理拓展版**，解决m_i不互质的问题（当然，当m_i互质，这份代码也是可以过的）

  核心思想：**将两个不定方程合并为一个不定方程**，合并$n-1$次就可以将n个方程合并为一个方程



其中：

- $[m_1, m_2]$ 代表m_1, m_2的**最大公约数**
- (m_1, m_2) 代表m_1, m_2的**最小公倍数**

{% note info %} **C++和python取模问题**

- C++负数取模，还是负数，所以求a模m的正余数时，`(a % m + m) % m`
- python负数取模，为正数，所以求a模m的正余数时，`a % m`
- 同时也是`km + a` 的**最小正整数解** {% endnote %}

- 代码

```
'''
Author: NEFU AB-IN
Date: 2022-03-11 20:29:00
FilePath: \ACM\Acwing\204.py
LastEditTime: 2022-03-11 20:56:04
'''


def exgcd(a, b):
    global k1, k2
    if b == 0:
        k1, k2 = 1, 0
        return a
```

```python
    d = exgcd(b, a % b)
    k1, k2 = k2, k1
    k2 -= (a // b) * k1
    return d


n = int(input())

m1, a1 = map(int, input().split())
flag = 0
for i in range(n - 1):
    m2, a2 = map(int, input().split())
    k1, k2 = 0, 0
    d = exgcd(m1, m2)
    if (a2 - a1) % d:
        flag = 1
        break
    k1 *= (a2 - a1) // d
    # k1' = k1 + k * (m2 // d) , k取任意整数
    t = m2 // d
    k1 = k1 % t  # 取最小的k1
    # x = a + km
    a1 = k1 * m1 + a1
    m1 = m1 // d * m2

if flag:
    print(-1)
else:
    print(a1 % m1)  #x的最小正整数解
```