欧几里得算法及其扩展*

张晴川
qzha536@aucklanduni.ac.nz

May 21, 2020

1 问题

二元一次不定方程：给定非负整数 $a, b, c (1 \leq a, b, c \leq 10^{18})$，求一组 $ax + by = c$ 的整数解 (x, y)。

2 定义

- $a \mid b$ 表示 a 整除 b，即存在 k 满足 $b = ka$
- $\gcd(a, b)$ 表示 a, b 的最大公约数 (greatest common divisor)。值得一提的是，最大公约数的定义其实是被所有公约数整除的数，不过在只考虑非负整数的情况下，也就相当于最大的那个公约数了。

3 欧几里得算法（辗转相除法）

设 g 是 a 和 b 的最大公约数，那么 $ax + by$ 一定也是 g 的倍数（为什么?）。所以如果 c 不是 g 的倍数，直接返回无解。所以我们先考虑如何计算最大公约数 $\gcd(a, b)$：

当 $b = 0$ 的时候，$\gcd(a, b) = \gcd(a, 0) = a$。

而 $b \neq 0$ 的时候，我们用 $\gcd(a, b) = \gcd(a - b, b)$ 来减小问题规模。以下是证明，一共分为两步：

1. a, b 的公约数一定是 $a - b, b$ 的公约数，不会遗漏：

$$\begin{aligned} d \mid a, d \mid b &\implies a = a'd, b = b'd \\ &\implies (a - b) = (a' - b')d, b = b'd \\ &\implies d \mid (a - b), d \mid b \end{aligned}$$

*更多内容请访问：https://github.com/SamZhangQingChuan/Editorials

2. $a-b,b$ 的公约数一定是 a,b 的公约数，不会新增：

$$
\begin{aligned}
d \mid (a-b), d \mid b &\implies (a-b)=c'd, b=b'd \\
&\implies a=(c'+b')d, b=b'd \\
&\implies d \mid (a-b), d \mid b
\end{aligned}
$$

于是得到 $\gcd(a,b)=\gcd(a-b,b)$。注意到 $a \bmod b$ 相当于 a 减去若干次 b，所以 $\gcd(a,b)=\gcd(a \bmod b, b)$

代码

欧几里得算法求最大公约数：

```
using ll = long long;

ll GCD (ll a, ll b) {
    while (b != 0) {
        a %= b;
        swap (a, b);
    }
    return a;
}
```

4 扩展欧几里得算法

解的存在性

现在假设 c 是 $\gcd(a,b)$ 的倍数，那么问题转化为如何解 $ax+by=\gcd(a,b)$，然后给解乘上 $\frac{c}{\gcd(a,b)}$ 即可。

具体流程

假设存在两个等式：

$$
\begin{aligned}
ax_1+by_1&=c_1\\
ax_2+by_2&=c_2
\end{aligned}
$$

可以得到：

$$a(x1-x_2)+b(y_1-y_2)=c_1-c_2$$

所以等式也可以像整数一样加减乘除。我们只需要以下两个方程开始，不断辗转相除直到等式右边是 $\gcd(a, b)$ 即可。

$$
\begin{aligned}
1a + 0b &= a \\
0a + 1b &= b
\end{aligned}
$$

具体参考以下代码。

代码

```cpp
using ll = long long;

tuple<ll, ll, ll> exGCD (ll a, ll b) {
    // 1. 初始化
    tuple<ll, ll, ll> equation[2] = {{1, 0, a},
                                     {0, 1, b}};
    int cnt = 0;
    while (get<2> (equation[1]) != 0) {
        // 2. 获取商
        ll q = get<2> (equation[0]) / get<2> (equation[1]);
        // 3. 得到新的等式
        equation[0] = {
                get<0> (equation[0]) - q * get<0> (equation[1]),
                get<1> (equation[0]) - q * get<1> (equation[1]),
                get<2> (equation[0]) - q * get<2> (equation[1])
        };
        swap (equation[0], equation[1]);
    }
    return equation[0];
}
```

5 应用：乘法逆元

问题 给定一对互素的正整数 a 和 m，即 $\gcd(a, m) = 1$。求一个整数 a^{-1} 满足 $aa^{-1} \equiv 0 \mod m$。

解法 首先我们解不定方程 $ax + my = 1$。由于互素，一定存在合法解。现在等式左边等于 1，所以等式左边等于 1，所以在模 m 意义下一定也等于 1。由于 my 是 m 的倍数，所以在模 m 意义下等于 0。那么就可以得到 $ax \equiv 1 \mod m$。让 $a^{-1} = x$ 就是所求的逆元了。