整除与算法设计报告

1. 整数与整除算法

(1) 整除的概念与性质

假设 a、b 是任意给定两个整数且 $a \neq 0$，若存在整数 q 满足 $b = a \times q$，则称 b 能被 a 整除或 a 能整除 b，记为 $a \mid b$，此时称 a 是 b 的因子或 b 是 a 的倍数。

对任意给定的整数 a，它至少能被 ± 1 或 $\pm a$ 整除，这四个因子也被称为 a 的平凡因子，a 的其他因子被称为非平凡因子。

(2) 素数与合数

对于任意给定的非零整数 a，若 $a \neq \pm 1$ 且没有非平凡因子，则称 a 为素数或质数，否则为合数。

(3) 最大公约数与最小公倍数

由于自然数 1 可整除任意整数，故对任意两个整数而言，它们的公因数总是存在，但通常会考察它们的最大公因数，表示 $gcd(a,b)$ 同样相乘就是它们的公倍数，通常也会考察最小公倍数表示为 $lcm(a,b)$ 规定 $gcd(0,0)=0$

(4) 带余除法

举例如 $16 \div 5 = 3 \cdots 1$

假设 a、b 是任意整数，若有 $b = a \times q + r$，$0 \leq r < |a|$，则必有 $gcd(b,a) = gcd(a,r)$

(5) 辗转相除法

假设 a 和 b 是任意整数，则必存在两个整数 k_1 和 k_2，满足 $gcd(a,b) = k_1 b + k_2 a$

记 $b = r_0$，$a = r_1$ 根据辗转相除法有 $r(i) = r(i+1) \times q(i+1) + r(i+2)$

直到 $r(n-1) = r(n) \times q(n+1)$ 其中 $gcd(b,a) = r(n)$

2、同余算术及其应用

(1) 同余关系

定义：对给定的正整数 m，任何整数按相对他们除以 m 所得余数是否相同，可分为 m 类（余数相同分为一类）

假设 m 是任意正整数，对两个整数 a 和 b，它们除以 m 以后得到的余数记为 $a \pmod m$ 和 $b \pmod m$，如果两个余数相等，即 $a \pmod m = b \pmod m$ 则称 a, b 为模 m 同余关系，记为 $a \equiv b \pmod m$，$b \equiv a \pmod m$

(2) 同余关系判定

定理：假设 m 是任一正整数，任意整数 a 和 b 具有模 m 同余关系，当且仅当 $a-b$ 能整除 m

(3) 同余的保加性和保乘性

定理：设 m 是任一正整数，若 a 与 b 模 m 同余，c 与 d 模 m 同余，a, b, c, d 整数，那么它们和与和也同余，即 $a \equiv b \pmod m$，$c \equiv d \pmod m$ 那么 $a+c \equiv (b+d) \pmod m$，$ac \equiv bd \pmod m$

定理：若 ac 与 b 模 m 同余且 c 与 m 互素，则有 a 与 b 模 m 同余

(4) 同余类的加法与乘法

假设 m 是任一正整数，对任意的整数的和 b，定义 $[a]_m + [b]_m = [a+b]_m$，$[a]_m \times [b]_m = [a \times b]_m$

(5) 同余方程组

假设 $m_1, m_2 \cdots m_k$ 为 k 个整数，$a_1, a_2 \cdots a_k$ 为 k 个整数，由 k 个以 x 为未知数的同余方程联立而求解的组为同余方程组

$$\begin{cases} x \equiv a_1 \pmod{m_1} \\ x \equiv a_2 \pmod{m_2} \\ x \equiv a_3 \pmod{m_3} \end{cases}$$

中国余数定理：设 $m_1, m_2 \cdots m_k$ 是两两互素的正整数，那么同余方程有唯一的以 $M = m_1, m_2 \cdots m_k$ 为模的解

3、算法设计策略与应用

(1) 蛮力策略

穷举法或暴力法，就是用最原始方法解决问题，基本思路是以扫描的模式依次处理问题域的每个元素，并将所有结果枚举出来进而实现问题求解基本步骤 ① 找到枚举范围 ② 找到约束条件

(2) 贪心策略

贪心算法就是从问题某个初始解出发通过每一步构造当前状态的局部最优解方法，一步步向最优解逼近，最终得到全局最优解

(3) 递归策略

若一个算法通过把问题转为更小规模相同问题方式进行求解，则称为递归

(4) 回溯策略

回溯法是一种选优搜索法，按选优条件向前搜索以达到目标，但当搜索到某一步时发现原先选择不优或不达目标就退回重选，这种不通过就退回再走技术称为回溯法

(5) 动态规划是通过拆分问题，定义问题状态和状态之间关系，使问题能够以递推方式解决