

任意精度整数运算

假设一个 m 位十进制整数 $u = (a_{m-1}a_{m-2} \cdots a_3a_2a_1a_0)_{10}$, 其中 $0 \leq a_i < 10, 0 \leq i < m$. 则 u 可以用一个数组表示

a_0	a_1	a_2	\cdots	a_{m-2}	a_{m-1}
-------	-------	-------	----------	-----------	-----------

利用这种表示方法, 可以计算任意精度的整数加法、减法、乘法、除法。

以“九千七百三十五” $(9735)_{10}$ 和“五百六十七” $(567)_{10}$ 相加为例:

5	3	7	9	
7	6	5		
12	9	12	9	
2	0	3	0	1

最终结果是 $(10302)_{10}$.

以上所用的方法是小学阶段学过的长整数运算的经典方法的计算机表示. 类似地, 可以处理减法、乘法和除法. 除法有点特殊, 小伙伴们采用试商法¹.

¹经过优化之后, 无论是什么进制, 都可以做到最多试两次, 参见 Knuth 的 *The Art of Computer Programming*, 第二

1. 通过命令行参数获得两个正整数, 利用上面介绍的长整数运算方法, 计算两者之和.
2. 在文件 input1.txt 中, 第一行是一个正整数 N, 表示下面还有 N 行, 每一行代表一个整数 (可能是负数), 每个数的具体形式为一个可能的符号 (+ 或 -) 之后跟随若干个数字. 请计算这 N 个整数的和, 然后打印这个结果. 例如对于文件

3

100

+200

-356

程序的输出为

-56

3. 计算

$$\sum_{k=1}^{100} k!$$

4. 计算级数的前一百位有效数字

$$\sum_{k=0}^{\infty} \frac{1}{k!}$$

大规模整数乘法通常采用环上离散傅立叶变换的方法, 比较复杂, 比小学生的方法的效率要高很多, 请参考 Schönhage–Strassen algorithm.