NOIP 03.6 高精度计算 High Precision Computation

Dubos.

高精度除法比另外三种运算复杂，我们把思路呈现如下：

int A[205], B[205], C[205], R[205]; ⓪

被除数， 除数， 余数， 商.

R[4] R[3] R[2] R[1]
0 0 3 2

B[3] B[2] B[1] ① A[4] A[3] A[2] A[1] ①
1 2 8) 4 0 9 6

0 　　 $R[4] = C(4) / B = 0$, ②③

4 0 　　 $C(4) = C(4)\,\%\,B = 4$, ③ $C(3) = C(4)*10 + A[3] = 40$, ④

0 　　 $R[3] = C(3) / B = 0$,

4 0 9 　　 $C(3) = C(3)\,\%\,B = 40$, $C(2) = C(3)*10 + A[2] = 409$,

3 8 4 　　 $R[2] = C(2) / B = 0$,

2 5 6 　　 $C(2) = C(2)\,\%\,B = 25$, $C(1) = C(2)*10 + A[1] = 256$,

2 5 6 　　 $R[1] = C(1) / B = 2$.

0 　　 $C(0) = C(1)\,\%\,B = 0$.

⓪ 这里为了图示简洁，我们只用一个字母作数组名，在讲义中为了代码易读，我们分别用 num1、num2、temp、result 表示被除数、除数、余数、商。

① 与高精度加、减、乘法不同，除法时常更改和使用各个数组的有效长度，所以我们用各个数组的第0位存储有效长度值，从第1位开始存储大整数的个位、十位……

② 写成 $R[4] = A[4] / B$ 更容易理解，但下面我们会观察出：

$R[i] = C(i) / B$, $C(i) = C(i+1)*10 + A[i]$

这两个式子也适用于 R(4) 和 C(4)，为与下面格式一致，写成 $R(4) = C(4) / B$

③ 目前没有大整数除法，更没有大整数求余数，所以要用大整数减法实现，每个数位最多要减 9 次才能得到对应数位上的商。

④ 大整除乘以 10，加一位整数