

1-1 A+B Problem

1. 使用的数据结构和算法原理

使用的数据结构：数组

使用的算法：高精度压位算法

原理：将大整数拆分为几位一组的小整数，将小整数分别做乘法之后相加得到两个大整数的乘法结果。也就是说将大整数的乘法转化为两个多项式的乘法。

实现要点：

(1) 压位位数

估算每一位要计算的加法数量，与 long long 数据类型的范围对比，计算出最多的压位位数（8 位）

(2) 进位操作

在最高位进位的时候要增加该整数的长度。

(3) 输入输出优化

(4) 计算新的结果的时候清空之前的数组

2. 遇到的问题

(1) 尽管用了压位仍然超时，最后发现是对两个整数拆分成的多项式中的每一项进行了多次重复计算。最后通过一个表格将每一项都记录下来，避免了重复计算。

(2) 一开始计算进位的时候是每次计算完一个数都计算一次进位，极大地增加了时间复杂度。最后改进为把所有数都计算完毕后再进行进位操作。

(3) 在输出的时候将数转换为字符串出现了一些问题，出现了把数反向输出或忘记输出 0 等情况

3. 时间与空间复杂度的估算

(1) 时间复杂度：

输入： $O(n)$

将每个输入拆分为 8 位一组的多项式： $O(n(l_a + l_b))$

计算乘法： $O(nl_al_b)$

进位： $O(l_a + l_b)$

输出： $O(n)$

总时间复杂度： $O(nl_al_b)$

(2) 空间复杂度：

输入： $O(n(l_a + l_b))$

中间处理： $O(l_a + l_b)$

输出： $O(n(l_a + l_b))$

总空间复杂度： $O(n(l_a + l_b))$