

苏州大学实验报告

院、系	计算机学院	年级专业	20 计科	姓名	柯骅	学号	2027405033
课程名称	数字电路与数字系统设计					成绩	

一、题目再现

设计一个程序实现大整数的加、减和乘。

二、解题与设计思路

1. 对于加法，利用小学时的加法原理，从个位数开始依次向高位开始，将两个数对应的数、进位(jw)加在一起，如果出现相加结果 >9 ，则 $jw=1$ ，否则 $jw=0$ ，将所得结果模 10 存入对应位置。

2. 对于减法，类比加法。

3. 对于乘法，

(1) 考虑一位数乘大整数，类比于加法，jw 可以 >1 。

(2) 从个位数开始枚举第一个数 x 的每一位数，将它与第二个数 y 分别相乘，存放在临时变量 `temp` 中，并累加到 `ans` 中。
最后累加所得 `ans` 即为乘法结果。

三、程序处理

1. 设计 `big_int` 类型，用于存放与计算大整数，有以下函数：

(1) `void build(string &s);` 用于将字符串 s 变成存在数组 a 里的大整数

(2) `void show();` 输出大整数

(3) `big_int operator+;` 计算两个大整数相加，返回 `big_int` 类型的结果

(4) `big_int operator-;` 计算两个大整数相减，返回 `big_int` 类型的结果

(5) `big_int operator*;` 计算两个大整数相乘，返回 `big_int` 类型的结果

2. 符号问题：

(1) 两个符号相同时，加法用 `operator+`，减法用 `operator-`，乘法结果符号为+

(2) 两个符号不同时，加法用 `operator-`，减法用 `operator+`，乘法结果符号为-

3. 读入正数时符号位可能消失，需要特判。

四、时空复杂度分析

1. 时间复杂度

(1) `build`: $O(n)$

(2) `operator+`: $O(n)$

(3) `operator-`: $O(n)$

(4) `operator*`: $O(n^2)$

(5) `show`: $O(n)$

总结：有乘法为 $O(n^2)$ ，无乘法为 $O(n)$

2. 空间复杂度

- (1) 存放整数的数组: $O(n)$
- (2) 存放加减结果的数组: $O(n)$
- (3) 存放乘法结果的数组: $O(2*n)$

总结: 数量级为 $O(n)$

五、运行结果

1. 正正

```
x=6156163
y=1651894321864
x+y=1651900478027
x-y=-1651888165701
x*y=10169330704169247832
```

```
6156163 + 1651894321864 =
1,651,900,478,027

6156163 - 1651894321864 =
-1,651,888,165,701

6156163 × 1651894321864 =
10,169,330,704,169,247,832
```

2. 正负

```
x=1561651
y=-46551651
x+y=-44990000
x-y=48113302
x*y=-72697432335801
```

```
1561651 + -46551651 =
-44,990,000

1561651 - -46551651 =
48,113,302

1561651 × -46551651 =
-72,697,432,335,801
```

3. 负正

```
x=-15489494
y=12318974653
x+y=12303485159
x-y=-12334464147
x*y=-190814683973795582
```

```
-15489494 + 12318974653 =
12,303,485,159

-15489494 - 12318974653 =
-12,334,464,147

-15489494 × 12318974653 =
-190,814,683,973,795,582
```

4. 负负

```
x=-1216544
y=-78943217
x+y=-80159761
x-y=77726673
x*y=96037896982048
```

```
-1216544 + -78943217 =
-80,159,761

-1216544 - -78943217 =
77,726,673

-1216544 × -78943217 =
96,037,896,982,048
```