

电子科技大学

计算机专业类课程

实验报告

课程名称：数据结构与算法

学院专业：计算机科学与工程学院

学生姓名：韩博宇

学号：2019040708023

指导教师：戴波

日期：2020 年 12 月 27 日

电子科技大学

实验报告

实验一

一、实验室名称：

电子科技大学清水河校区主楼 A2-412

二、实验项目名称：

线性表实验

三、实验原理：

利用线性数据结构实现长整数的存储和操作

四、实验目的：

练习使用线性表解决问题

五、实验内容：

生成大数据的四则运算器并根据输入得到计算结果

六、实验器材（设备、元器件）：

硬件平台：

CPU: Intel (R) Core (TM) i7-9750H CPU @ 2.60GHz

内存: 16G

软件平台：

操作系统: Windows 10

开发环境: Dev-C++ 5.11

七、实验步骤:

1. 问题描述

生成大数据的四则运算器并根据输入得到计算结果

2. 算法分析与概要设计

输入：两个大整数以及相应的运算符对应序号

输出：两个大整数运算结果

文字算法描述:

(1) 两个大整数的加法:

存储结构: 单链表

算法描述:

将输入的两个大整数分别存入单链表 `link_one`, `link_two` 中, `link_one` 长度记为 `n`, `link_two` 长度记为 `m`。

①从头结点开始, 依次访问两个链表中结点存储的整型数字, 进位 `temp` 设置为 0, 将两个数字以及进位 `temp` 相加, 相加结果 `mod10` 赋值给 `temp`, 相加结果整除 10 存入新的链表 `link_three`。两个单链表指针指向下一个结点;

②重复步骤①, 直至有一个链表或两个链表遍历完毕;

③若两个单链表都遍历完成, 若 `temp` 不为 0, 则将 `temp` 对应整数插入 `link_three`; 若其中一个链表遍历完整, 依次遍历另一个未遍历完全的链表, 结点存储的整型数字与 `temp` 相加, 重复步骤二中的赋值存储操作, 直至链表遍历完整;

④从头结点开始, 依次输出 `link_three` 中各个结点存储的整型

(2) 两个大整数的减法:

存储结构: 单链表

算法描述:

将输入的两个大整数分别存入单链表 `link_one`, `link_two` 中, `link_one` 长度记为 `n`, `link_two` 长度记为 `m`。

①若 $n \neq m$, `Node* p` 指向两个链表长度更大的头结点, `Node* q` 指向较小者。若 `p->data < q->data`, 则需要借位, 将 `p->data+10-q->data` 赋值

给 temp, $p \rightarrow next \rightarrow data$ 减一; 若 $p \rightarrow data \geq q \rightarrow data$, 将 $p \rightarrow data - q \rightarrow data$ 赋值给 temp。将 temp 值插入 link_three, p 和 q 依次指向下一结点, 直至 q 所指链表遍历完毕;

②将 $p \rightarrow data$ 值插入 link_three, p 指向下一结点, 直至 p 所指链表遍历完毕;

③若 $n < m$, 将 -1 插入 link_three; 若 $n > m$ 则不进行操作;

④若 $n == m$, 重复步骤二, Node* p 指向 link_one 头结点, Node* q 指向 link_two 头结点。但结束条件改变为遍历完 $m-1$ 个结点。最终判断两个链表最后一个结点的数值大小。若 $p \rightarrow data \geq q \rightarrow data$, 将 $p \rightarrow data - q \rightarrow data$ 赋值给 temp, 将 temp 值插入 link_three。反之将 $q \rightarrow data - p \rightarrow data$ 赋值给 temp, 将 temp 值插入 link_three, 再将 -1 插入 link_three;

⑤从 link_three 头结点指向的第一个结点开始输出, 若第一个结点为 -1, 则输出 ' - ' ; 若是正数则输出该值, 直至所有结点输出完毕。

(3) 两个大整数的乘法:

存储结构: 数组

算法描述:

将输入的两个大整数存入 string s1, string s2。初始化结果存储数组 num (置 0), $m = s1.size, n = s2.size$ 。

①将 s1, s2 分别逆序排列;

② $s1[i]$ 为 string s1 中第 $i+1$ 位数字, $s2[j]$ 表示 string s2 中第 $j+1$ 位数字。将 $(s1[i] - '0') * (s2[j] - '0')$ 赋值给 temp, $num[i+j] += temp \% 10$, 然后处理进位, $num[i+j] += temp / 10$, 当 $i = m-1, j = n-1$ 时退出循环;

③处理结果数组里的进位, 输出结果数组。

(4) 两个大整数的除法:

存储结构: 数组

算法描述:

①截数。从被除数的最高位起, 除数是几位数就从左边截出几位数, 当被截出的数小于除数时, 应再截一位数。

②试商。用 1-9 中的适当数字作为初商, 用初商去乘除数, 使所得的积小于(或等于)所截取的数, 并从截取的数中减去这个积, 所得差应小于除数, 差也可能是零。

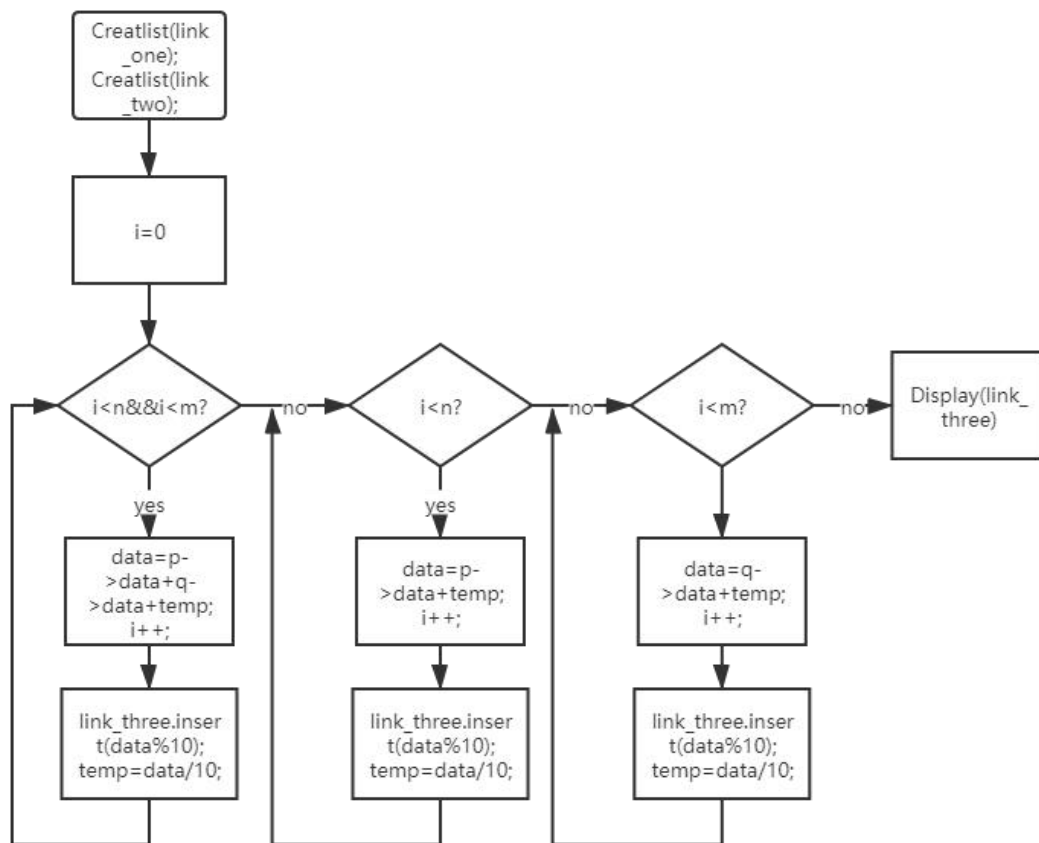
③再截数。将被除数第一次被截后余下的数, 紧接着写在差的后面, 称为第一余数, 从第一余数中第二次截数, 所截位数仍与除数的位数相

同，当第二次被截数小于除数时，应再截一位数。

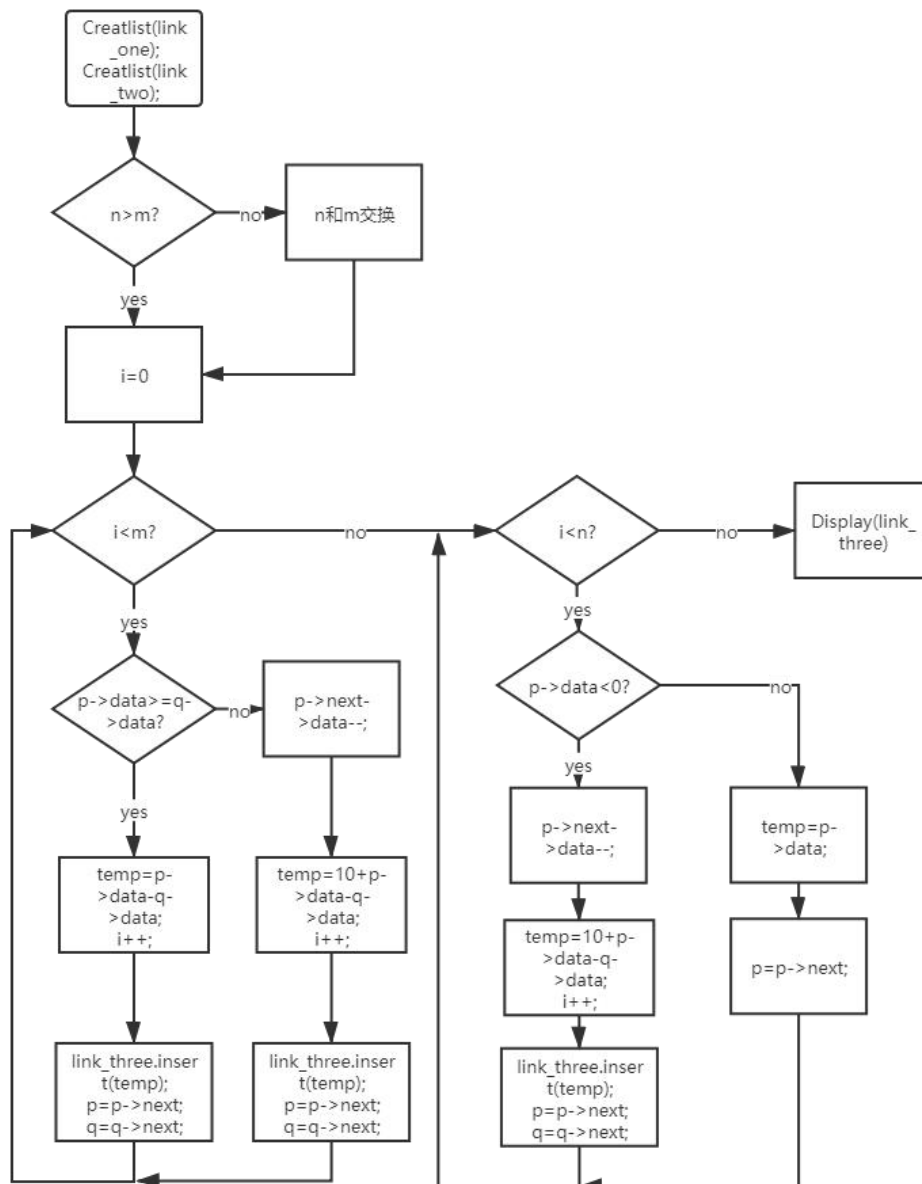
④再试商。仍用 1-9 中的适当数字作为次商，用次商去乘除数，使所得的积小于(或等于)第二次截得的数，并从第二次截取的数中减去这个积，所得差应小于除数，差也可能是零，将被除数第二次被截后余下的数，紧接着写在第二次差的后面，称为第二次余数。

⑤初商应写在第一次被截数的最末一位数字上边，次商应写在第二次被截数的最末一位数字上边，如初商和次商之间有空位应补 0, 0 的个数与空位的个数相同。

⑥重复上述步骤直到被除数的个位数字被截下参与计算完为止。如果最后一次差为 0，把各次所得商按先后顺序从左到右排好，同时输出余数。



大整数的加法流程图



大整数的减法流程图

八、实验数据及结果分析：

两个大整数的加法测试

```

C:\Users\86138\Desktop\数据结构与算法_大实验\实验1\text1.exe
请输入一个运算式，数字和符号之间用空格隔开，例如：1（空格）*（空格）2
18474874874874784 + 494848549848494944
18474874874874784+494848549848494944=513323424723369728

-----
Process exited after 11 seconds with return value 0
请按任意键继续. . .
  
```

两个大整数的减法测试

C:\Users\86138\Desktop\数据结构与算法_大实验\实验1\text1.exe

```
请输入一个运算式，数字和符号之间用空格隔开，例如：1（空格）*（空格）2
49498494849494949 - 4848848488878888
49498494849494949-4848848488878888=44649646360616061
```

```
-----
Process exited after 20.93 seconds with return value 0
请按任意键继续. . .
```

C:\Users\86138\Desktop\数据结构与算法_大实验\实验1\text1.exe

```
请输入一个运算式，数字和符号之间用空格隔开，例如：1（空格）*（空格）2
123456789 - 1123456789
123456789-1123456789=-10000000000
```

```
-----
Process exited after 26.35 seconds with return value 0
请按任意键继续. . .
```

C:\Users\86138\Desktop\数据结构与算法_大实验\实验1\text1.exe

```
请输入一个运算式，数字和符号之间用空格隔开，例如：1（空格）*（空格）2
548481584841548 - 548481584841548
548481584841548-548481584841548=0
```

```
-----
Process exited after 14.28 seconds with return value 0
请按任意键继续. . .
```

两个大整数的乘法测试

C:\Users\86138\Desktop\数据结构与算法_大实验\实验1\text1.exe

```
请输入一个运算式，数字和符号之间用空格隔开，例如：1（空格）*（空格）2
456456464 * 498494989878
456456464*498494989878=227541260401427671392
```

```
-----
Process exited after 13.86 seconds with return value 0
请按任意键继续. . .
```

两个大整数的除法测试

C:\Users\86138\Desktop\数据结构与算法_大实验\实验1\text1.exe

```
请输入一个运算式，数字和符号之间用空格隔开，例如：1（空格）*（空格）2
1644484894998789789 / 54848154564
1644484894998789789/54848154564=029982501 remainder is 45935505225
```

```
-----
Process exited after 14.57 seconds with return value 0
请按任意键继续. . .
```

由测试结果来看，大整数的四则运算都能通过程序实现并输出正确结果。

九、总结及心得体会：

在编写程序的过程中，既让我更加理解了线性表的操作，也进一步锻炼了我程序设计的能力。通过本次实验我进一步提升了自己分析和解决问题的能力，并且将所学运用到实际。本次实验的缺点在于：大整数的四则运算，并没有在原来的基础上加以改进，以减小时间和空间的复杂度；并且只能进行单步加减乘除运算，不能进行多步带括号的运算。

十、对本实验过程及方法、手段的改进建议及展望：

本实验的时间复杂度为 $O(n^2)$ ，在数值较大时运算时间很长，后期需要进行算法优化以减小时间和空间的复杂度，例如对于乘法我们可以使用最优子结构的性质进行优化。

报告评分：

指导教师签字：