

# 数据结构实验报告

## 实验一：一元多项式的加减

姓 名： 彭 思 翔

学 号： 2017301500061

班 级： 计科二班

日 期： 2018 年 10 月 21 日

上机环境： Win10 VSCode

## 一、实验题目

实验一：一元多项式的加减运算

### 【问题描述】

请编写一个程序，完成一元多项式的存储，并实现两个多项式的相加减及相乘运算。

### 【基本要求】

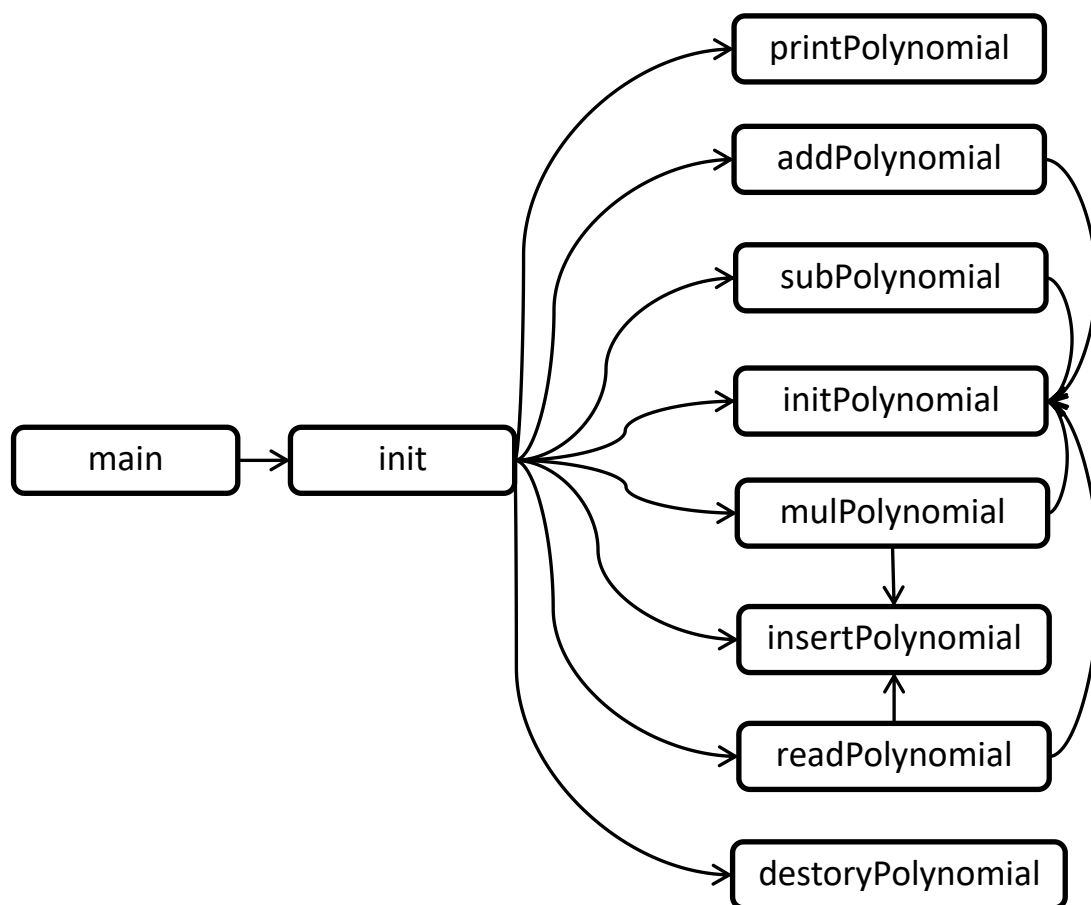
(1) 随机从键盘上输入多项式，构建单链表存储一元多项式，其中单链表的数据域包括系数和指数两项

(2) 最后的结果输出形式为  $\text{result}=3+4X^2+5X^2$

## 二、实验项目的目的

深入理解单链表并进行应用，顺便进行字符串处理，学会合理地设置函数，使函数复用率高。

## 三、实验项目程序结构



## 四、实验项目中各文件函数功能描述

```
void init(); //读入数据, 控制进程
void initPolynomial(Polynomial*&); //初始化多项式单链表
void readPolynomial(Polynomial*&); //读取多项式, 进行字符串处理
void printPolynomial(Polynomial*); //打印多项式
void destoryPolynomial(Polynomial*&); //销毁多项式单链表
void insertPolynomial(Polynomial*&, Polynomial*&); //插入排序单式至多项式的正确位置
void addPolynomial(Polynomial*&, Polynomial*&, Polynomial*&); //多项式加法, 由第三个参数返回结果
void subPolynomial(Polynomial*&, Polynomial*&, Polynomial*&); //多项式减法, 由第三个参数返回结果
void mulPolynomial(Polynomial*&, Polynomial*&, Polynomial*&); //多项式乘法, 由第三个参数返回结果
```

## 五、算法描述

### 【数据结构】

有序单链表: 用于存储和表示指数由高到低的多项式, 链表中存储指数、系数和下一项的地址, 只存储系数不为 0 的项。

```
typedef struct Poly {
    double coefficient, index;
    struct Poly * next;
}Polynomial;
```

### 【设计思路】

(1) 读取实现: 对于一个多项式的字符串, 以加号分割, 加号后第一个为数字则为系数, 否则系数为 1, 字符 'x' 后若为 '^' 则指数为其后面的数字, 否则指数为 1。对于数, 遇见连续数字在小数点前则前面的数 $\times 10$ 加上当前的数字, 在小数点后则前面的数+当前的数字/小数位数。

时间复杂度  $O(\text{Lenth})$

(2) 插入实现: 插入排序, 从前至后找到比需要插入单式的指数更小的位置前一个为止, 若指数相等则合并, 若指数不相等则插入。

时间复杂度  $O(n)$

```
void insertPolynomial(Polynomial *&L, Polynomial *&x) {
    Polynomial *p = L;
    while (p 下一个不为空 && p 下一个指数大于 x 的指数) p = p->next;
    if (p 与 x 指数相等 && p != L) 合并系数; else 插入 x 至 p 后;
}
```

(3) 加法实现: 两个多项式归并, 指数上若  $A[i] > B[j]$  则  $i++$ , 若  $A[i] < B[j]$  则  $j++$ , 若  $A[i] == B[j]$  则相加系数并  $i++j++$ 。

时间复杂度  $O(m+n)$

```
void addPolynomial(Polynomial *&A, Polynomial *&B, Polynomial *&C) {
    Polynomial *p = A->next, *q = B->next, *r = C, *s;
    while (p, q 不为 null) {
        if (p->index > q->index) 插入 p 至 C, p 后移;
```

```

        else if (p->index < q -> index) 插入 q 至 C, q 后移;
        else if (p->index == q -> index) 系数相加不为 0 则插入至 C, p 和 q 后移;
    }
    while (p 枚举剩余 A) p 加入 C;
    while (q 枚举剩余 B) q 加入 C;
}

```

(4) 减法实现：两个多项式归并，指数上若  $A[i] > B[j]$  则  $i++$ ，若  $A[i] < B[j]$  则  $j++$ ，若  $A[i] == B[j]$  则相减系数并  $i++j++$ 。代码和加法类似。

时间复杂度  $O(m+n)$

(5) 乘法实现：暴力枚举两个多项式的每一位，并相乘，将结果项插入至新的多项式。

时间复杂度  $O(m*n^2)$

```

void mulPolynomial(Polynomial *&A, Polynomial *&B, Polynomial *&C) {
    Polynomial *p = A->next, *q = B->next, *r = C, *s;
    while (p 枚举 A)
        while (q 枚举 B)
            系数相乘，指数相加，并插入 C;
}

```

## 六、实验数据和实验结果分析

相加：

```

Please enter the first polynomial:
x+x^3+x^4
Please enter the second polynomial:
0.123x^3-x+x^4
Please select a polynomial operation(1.Add;2.Subtract;3.Multiply):
2
Result:
0.877x^3+2x
请按任意键继续. . .

```

相减：

```

Please enter the first polynomial:
x^2-123.423x^3+x
Please enter the second polynomial:
-5x^3-0.232x-x^2
Please select a polynomial operation(1.Add;2.Subtract;3.Multiply):
1
Result:
-128.423x^3+0.768x
请按任意键继续. . .

```

相乘:

```
Please enter the first polynomial:
10.22x^100-x^1000+0.24x^233
Please enter the second polynomial:
78x^256+890.02x^3-x
Please select a polynomial operation(1.Add;2.Subtract;3.Multiply):
3
Result:
-78x^1256-890.02x^1003+x^1001+18.72x^489+797.16x^356+213.605x^236-0.24x^234+9096x^103-10.22x^101
请按任意键继续. . .
```

运行结果良好。

## 七、实验体会

对单链表有了更加深入的认识，熟悉了二路归并等相关算法。运算的实现倒还算是简单，处理输入的字符串却废了不少功夫，比如系数指数省略，负数、小数的处理等，看来对字符串的操作要更熟悉一点才行。另外乘法运算过于暴力，暂时未想出以当前数据结构的更优解决方案。