

湖南大学

数据结构

课程实验报告

题 目： 多项式加法

学生姓名 吴多智

学生学号 201626010520

专业班级 软件 1605

完成日期 2017.11.3

一、需求分析

1.问题描述

设 $P_n(x)$ 和 $Q_m(x)$ 分别两个一元多项式。试编写程序实现一元多项式的加法运算。设 $P_n(x) = 9x^{15} + 6x^8$, $Q_m(x) = 8x^{15} + 9x^2$, 则两者相加的结果为 $17x^{15} + 6x^8 + 9x^2$ 。

2.输入输出数据

//第一个多项式为 $9x^{15} + 7x^8 + 5x^3 + 3x$

输入

4 //表示第一个多项式的项数

9, 15 (回车) //表示 $9x^{15}$

7, 8 (回车)

5, 3 (回车)

3, 1 (回车)

输出

$9x^{15} + 7x^8 + 5x^3 + 3x^1$

//第二个多项式为 $-7x^8 + 6x^3 + 2$

输入

3 //表示第二个多项式的项数

6, 3 (回车)

-7, 8 (回车) //表示 $-7x^8$

2, 0 (回车)

输出

$-7x^8 + 6x^3 + 2x^0$

求和结果

$9x^{15} + 11x^3 + 3x^1 + 2x^0$

3.测试样例设计

3.1 正常测试

输入

4 //表示第一个多项式的项数

9, 15 (回车) //表示 $9x^{15}$

7, 8 (回车)

5, 3 (回车)

3, 1 (回车)

输出

$9x^{15} + 7x^8 + 5x^3 + 3x^1$

输入

3 //表示第二个多项式的项数

6, 3 (回车)

-7, 8 (回车) //表示 $-7x^8$

2, 0 (回车)

输出

$-7x^8 + 6x^3 + 2x^0$

求和结果

$9x^{15} + 11x^3 + 3x^1 + 2x^0$

3.2 两个多项式没有相同的项

输入

4 //表示第一个多项式的项数
 9, 15 (回车) //表示 $9x^{15}$
 7, 14 (回车)
 5, 13 (回车)
 3, 11 (回车)

输出

$9x^{15} + 7x^{14} + 5x^{13} + 3x^{11}$

输入

3 //表示第二个多项式的项数
 6, 3 (回车)
 -7, 8 (回车) //表示 $-7x^8$
 2, 0 (回车)

输出

$-7x^8 + 6x^3 + 2x^0$

求和结果

$9x^{15} + 7x^{14} + 5x^{13} + 3x^{11} - 7x^8 + 6x^3 + 2x^0$

3.3 两个相同的多项式

输入

4 //表示第一个多项式的项数
 9, 15 (回车) //表示 $9x^{15}$
 7, 14 (回车)
 5, 13 (回车)
 3, 11 (回车)

输出

$9x^{15} + 7x^{14} + 5x^{13} + 3x^{11}$

输入

4 //表示第二个多项式的项数
 9, 15 (回车) //表示 $9x^{15}$
 7, 14 (回车)
 5, 13 (回车)
 3, 11 (回车)

输出

$9x^{15} + 7x^{14} + 5x^{13} + 3x^{11}$

求和结果

$18x^{15} + 14x^{14} + 10x^{13} + 6x^{11}$

3.4 测试两个都是比较长的多项式

输入

8 //表示第一个多项式的项数
 9, 15 (回车) //表示 $9x^{15}$
 7, 14 (回车)
 5, 13 (回车)
 3, 11 (回车)
 -1 9 (回车)

```

-3 20 (回车)
20 2 (回车)
11 9 (回车)
输出
-3x^20+9x^15+ 7x^14+5x^13+3x^11+10x^9+20x^2
输入
7 //表示第二个多项式的项数
11 9 (回车)
11 45 (回车)
8 8 (回车)
11 20 (回车)
4 11 (回车)
8 8 (回车)
9 9 (回车)
输出
11x^45+11x^20+4x^11+20x^9+16x^8
求和结果
11x^45+8x^20+9x^15+ 7x^14+5x^13+7x^11+30x^9+16x^8+20x^2
3.5 测试输入的单项式都是一样的
输入
5 //表示第一个多项式的项数
9, 9 (回车)
9, 9 (回车)
9, 9 (回车)
9, 9 (回车)
9, 9 (回车)
输出
45x^9
输入
5 //表示第二个多项式的项数
10, 11 (回车)
10, 11 (回车)
10, 11 (回车)
10, 11 (回车)
10, 11 (回车)
输出
50x^11
求和结果
50x^11+45x^9

```

二、概要设计

(这个部分一般要撰写三个方面的内容)

1.抽象数据类型

实验要求完成多项式的加法，所以先有个项的节点类。由于多项式在输入时，并非按幂的大小顺序来输入，但又要时刻维持线性表中的节点是按照幂的大小次序排列的，所以

线性表的插入操作比较频繁，因此选用链表实现线性表。

节点类：

```
class Node{
public:
    int s;
    int u;
    Node* next;
    Node(int a,int b){
        s = a;
        u = b;
        next = NULL;
    }
    Node(){
        next = NULL;
    }
};
```

线性表 ADT:

```
class Link{
private:
    Node* head;
    int size;
public:
    Link(){
        head = NULL;
        size = 0;
    }

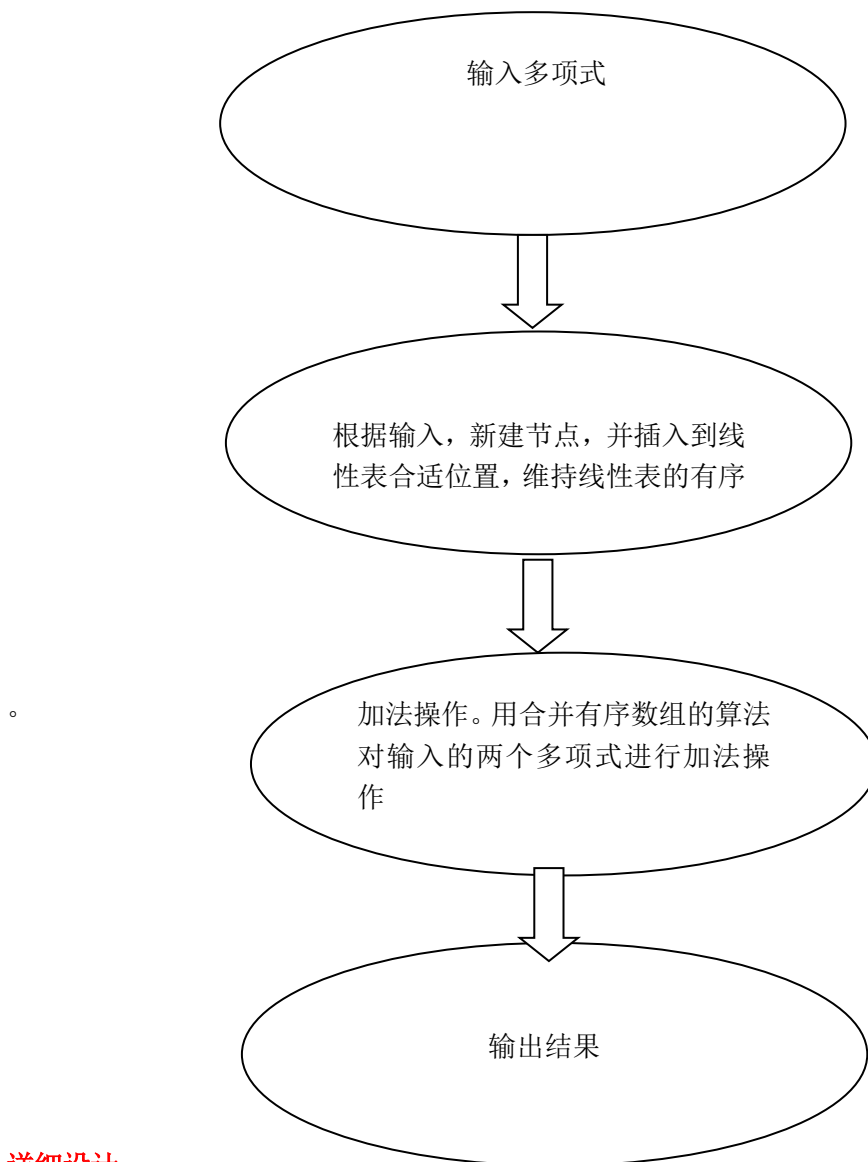
    void add(Link l); //传入一个多项式，实现相加，结果存在当前链表

    void add(int s,int u); //添加节点
    void insert(Node* n); //插入节点
    void print(); //打印多项式
    int length(){return size;};
    Node* Head(){return head;}; //返回头结点
};
```

2.算法的基本思想

实验是进行二项式的加法，所以线性表的主要操作的多项式的插入与加法。插入操作中，需要定位插入位置，然后插入，做法与上一实验单链表的插入一样；在多项式的加法操作中，由于两个多项式都是有序的，也就相当于合并两个有序的数组，是合并后的数组依然有序，因此多项式加法使用同样的算法。

3.程序的流程



三、详细设计

(这个部分一般要撰写四个方面的内容)

1.物理数据类型

对于多项式的每个项，使用节点类存储；整个多项式，使用基于链表的线性表来存储，同时，添加节点是要维持线性表的有序。

2.输入和输出的格式

输入

5 (回车) //表示第一个多项式的项数
 9, 9 (回车)
 9, 9 (回车)
 9, 9 (回车)
 9, 9 (回车)
 9, 9 (回车)

输出

45x^9

输入

```
5      (回车) //表示第二个多项式的项数
10, 11 (回车)
10, 11 (回车)
10, 11 (回车)
10, 11 (回车)
10, 11 (回车)
10, 11 (回车)
```

输出

$50x^{11}$

求和结果

$50x^{11}+45x^9$

3.算法的具体步骤

本程序所实现的线性表主要用于解决实验中的问题，只要操作是多项式的添加与加法。

3.1 多项式的添加

```
void Link::insert(Node* n){

    Node * cur = head;
    Node* par = head;
    while(cur!=NULL){
        if(n->u >= cur->u){
            break;
        }
        par = cur;
        cur = cur->next;
    }

    if(cur == head){ //头部插入
        if(cur->u == n->u){
            head->s +=n->s; //当系数为 0 时
            if(head->s == 0){
                if(head->next == NULL){
                    head = NULL;
                }else{
                    Node* t = head;
                    head = head->next;
                    delete t;
                }
            }
        }else{
            n->next = head;
            head = n;
        }
    }else if(cur == NULL){ //尾部插入
```

```

        par->next = n;
    }else{
        if(cur->u == n->u){
            cur->s += n->s;
            if(cur->s == 0){    //当系数相等时
                Node* t = cur;
                par->next = cur->next;
                delete t;
            }
        }else{
            n->next = cur;
            par->next = n;
        }
    }
}

```

3.2 多项式加法

```

Link Link::add(Link l){
    Link r = Link();
    const Node* t = l.Head();
    Node* n = head;
    while(t->next!=NULL && n->next!=NULL){
        if(t->u == n->u){
            int s = t->s+n->s;
            if(s==0){
                t = t->next;
                n = n->next;
                continue;
            }

            r.add(s,t->u);
            t = t->next;
            n = n->next;
        }else if(t->u > n->u){
            r.add(t->s,t->u);
            t = t->next;
        }else{
            r.add(n->s,n->u);
            n = n->next;
        }
    }

    while(t!=NULL){
        r.add(t->s,t->u);
        t = t->next;
    }
}

```



```

    }

    while(n!=NULL){
        r.add(n->s,n->u);
        n = n->next;
    }

    return r;
}

```

4.算法的时空分析

在添加的操作中，由于有保持线性表的有序，需要定位插入位置，时间复杂度为 $\theta(n)$ ，由于底层为链表实现，插入操作时间复杂度为 $\theta(1)$ ，总的时间复杂度为 $\theta(n)$ ；多项式加法操作中，只需要同时遍历两线性表一次，时间复杂度为 $\theta(n+m)$ ，故总的时间复杂度为 $\theta(n)$ 。

四、调试分析

(分两个阶段完成。)

1.调试方案设计

本实验基于链表来实现线性表，再插入过程中，处理不当会有空指针出现，故需要调试观察变量的值是否与预期相同；同时，相加阶段也是如此，都需要通过调试看程序是如何一步步运行的，这样能更快定位问题。

【以上部分在预备实验报告就需完成】

五、测试结果

```

H:\数据结构\HomeWork\core\sy02.exe
请输入第一个多项式:
4
9 15
7 8
5 3
3 1
您输入的第一个多项式: 9x^15+7x^8+5x^3+3x^1
请输入第二个多项式:
3
6 3
-7 8
2 0
您输入的第二个多项式: -7x^8+6x^3+2x^0
求和结果: 9x^15+11x^3+3x^1+2x^0

```

```

H:\数据结构\HomeWork\core\sy02.exe
请输入第一个多项式:
4
9 15
7 14
5 13
3 11
您输入的第一个多项式: 9x^15+7x^14+5x^13+3x^11
请输入第二个多项式:
3
6 3
-7 8
2 0
您输入的第二个多项式: -7x^8+6x^3+2x^0
求和结果: 9x^15+7x^14+5x^13+3x^11-7x^8+6x^3+2x^0_

```

```

H:\数据结构\HomeWork\core\sy02.exe
请输入第一个多项式:
4
9 15
7 14
5 13
3 11
您输入的第一个多项式: 9x^15+7x^14+5x^13+3x^11
请输入第二个多项式:
4
9 15
7 14
5 13
3 11
您输入的第二个多项式: 9x^15+7x^14+5x^13+3x^11
求和结果: 18x^15+14x^14+10x^13+6x^11_

```

```

H:\数据结构\HomeWork\core\sy02.exe
请输入第一个多项式:
8
9 15
7 14
5 13
3 11
-1 9
-3 20
20 2
11 9
您输入的第一个多项式:  $-3x^{20}+9x^{15}+7x^{14}+5x^{13}+3x^{11}+10x^9+20x^2$ 
请输入第二个多项式:
7
11 9
11 45
8 8
11 20
4 11
8 8
9 9
您输入的第二个多项式:  $11x^{45}+11x^{20}+4x^{11}+20x^9+16x^8$ 
求和结果:  $11x^{45}+8x^{20}+9x^{15}+7x^{14}+5x^{13}+7x^{11}+30x^9+16x^8+20x^2$ 

```

```

H:\数据结构\HomeWork\core\sy02.exe
请输入第一个多项式:
5
9 9
9 9
9 9
9 9
9 9
您输入的第一个多项式:  $45x^9$ 
请输入第二个多项式:
5
10 11
10 11
10 11
10 11
10 11
您输入的第二个多项式:  $50x^{11}$ 
求和结果:  $50x^{11}+45x^9$ 

```

六、实验心得

在实验中，加深了对指针的理解与运用，同时，实验是用链表实现的有序线性表，有很多的插入操作，在没有学习数据结构之前，自己都是用的数组存数据，要插入时，总要把插入之后的数据往后移动一位，但链表不一样，只需要改变指针的指向，就能完成插入操作，时间复杂度为常数。同时，多项式相加时，要考虑系数为零时删除节点的情况，并且还要考虑要当前的节点是否为头结点，这让我深刻体会到维护链表头结点的重要性，不论是删除、插入节点，都要考虑头结点。