2.5 线性表的应用-两个多项式相加

2.5.1 问题描述

多项式
$$p(x) = c_1 x^{e_1} + c_2 x^{e_2} + \dots + c_m x^{e_m}$$

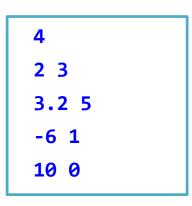
求两个多项式相加的程序

例如,
$$p(x)=2x^3+3.2x^5-6x+10$$
, $q(x)=6x+1.8x^5-2x^3+x^2-2.5x^4-5$
$$r(x)=p(x)+q(x)$$

$$r(x)=5x^5-2.5x^4+x^2+5$$

两个多项式的数据分别存放在abc1.in和abc2.in文本文件中,要求相加的结 果多项式的数据存放在abc.out文本文件中。

abc1.in文件



$$p(x)=2x^3+3.2x^5-6x+10$$

abc2.in文件

$$q(x)=6x+1.8x^5-2x^3+x^2-2.5x^4-5$$

abc.out文件



第1个多项式: [[2.0, 3], [3.2, 5], [-6.0, 1], [10.0, 0]]排序后结果: [[3.2, 5], [2.0, 3], [-6.0, 1], [10.0, 0]]第2个多项式: [[6.0, 1], [1.8, 5], [-2.0, 3], [1.0, 2], [-2.5, 4], [-5.0, 0]]排序后结果: [[1.8, 5], [-2.5, 4], [-2.0, 3], [1.0, 2], [6.0, 1], [-5.0, 0]]相加多项式: [[5.0, 5], [-2.5, 4], [1.0, 2], [5.0, 0]]

```
#多项式抽象数据类型
ADT PolyClass
数据对象:
    PolyElem=\{(c_i, e_i) \mid 1 \le i \le n, c_i \in \text{float}, e_i \in \text{int}\};
数据关系:
    r=\{\langle x_i, y_i \rangle \mid x_i, y_i \in PolyElem, i=1, \dots, n-1\}
基本运算:
    Add(e):将多项式项e添加到末尾。
    CreateList(fname): 从fname文件中读取数据建立多项式。
    getsize(): 返回多项式的项数
    getitem(i): 返回序号为i的多项式项。
    getdata(): 返回多项式。
    Sort():对多项式按指数递减排序。
    PolyAdd(B): 返回当前多项式与多项式B的相加结果。
} #ADT PolyClass
```

2.5.2 问题求解

1. 设计顺序存储结构

- 多项式的每一项用一个列表[c_i , e_i](其中 c_i 为系数, e_i 为指数) 存储,一个多项式顺序表用元素为列表[c_i , e_i]的列表data存储。
- 例如,多项式 $p(x)=2x^3+3.2x^5-6x+10$ 的data列表为[[2.0, 3], [3.2, 5], [-6.0, 1], [10.0, 0]]。

多项式顺序表类PolyList

2. 设计PolyList的基本运算算法

(1) PolyList的构造方法

```
def __init__(self):#构造方法self.data=[]#存放多项式项的列表
```

(2) 将多项式项e添加到末尾Add(e)

```
def Add(self,e): #添加一个多项式项e
self.data.append(e)
```

(3) 创建多项式顺序表CreateList(fname)

```
def CreateList(self,fname): #从fname文件中读取多项式数据并添加到data
fin=open(fname,"r")
n=int(fin.readline().strip())
for i in range(n):
    p=fin.readline().strip().split()
    self.data.append([float(p[0]),int(p[1])])
fin.close()
```

(4) 返回多项式的项数getsize()

def getsize(self): #求多项式的项数 return len(self.data)

(5) 返回序号为i的多项式项

```
def __getitem__(self,i): #求序号为i的元素 return self.data[i]
```

(6) 返回多项式的data列表getdata()

def getdata(self): #返回多项式列表 return self.data

(7) 对多项式接指数递减排序Sort()

```
def Sort(self): #对data按指数递减排序
self.data=sorted(self.data,key=itemgetter(1),reverse=True)
```

(8) 返回当前多项式与多项式B的相加结果PolyAdd(B)

两个按指数递减排序的多项式顺序表



相加的结果多项式顺序表

用i、j分别遍历A和B中的元素,先建立一个空多项式顺序表C,在i、j都没有遍历完时循环,取i指向的A中元素p,取j指向的B中元素q:

- ① 若p元素的指数 (p[1]) 较大,将p元素添加到C中,i增加1。
- ② 若q元素的指数 (q[1]) 较大,将q元素添加到C中,j增加1。
- ③ 此时p、q元素的指数相同(p[1]=q[1]),求出它们的系数和k(k=p[0]+q[0]),如果 $k\neq 0$,由k和p[1]新建一个元素并添加到C中,否则不新建结点,并将i、j均增加1。

上述循环过程结束后,若有一个多项式顺序表没有遍历完,说明余下的多项式项都是指数较小的多项式项,将它们均添加到C中,最后返回C。

```
#当前多项式和多项式B的相加运算
def PolyAdd(self,B):
                            #新建结果多项式顺序表
 C=PolyList()
                            #多项式A的项数
 m=len(self.data)
                            #多项式B的项数
 n=B.getsize()
 i, j=0,0
 while i<m and j<n:
   p,q=self.data[i],B[j]
                            #将较大指数的p项添加到C中
   if p[1]>q[1]:
     C.Add(p)
     i+=1
                            #将较大指数的q项添加到C中
   elif q[1]>p[1]:
     C.Add(q)
     j+=1
                            #两指数相同,即p[1]=q[1]
   else:
                            #系数相加为k
     k=p[0]+q[0]
                            #k不为Ø时添加相应项到C中
     if (k!=0):
      C.Add([k,p[1]])
     i+=1
     j+=1
```

```
      while i<m:</td>
      #将A余下的项添加到C中

      p=self.data[i]
      C.Add(p)

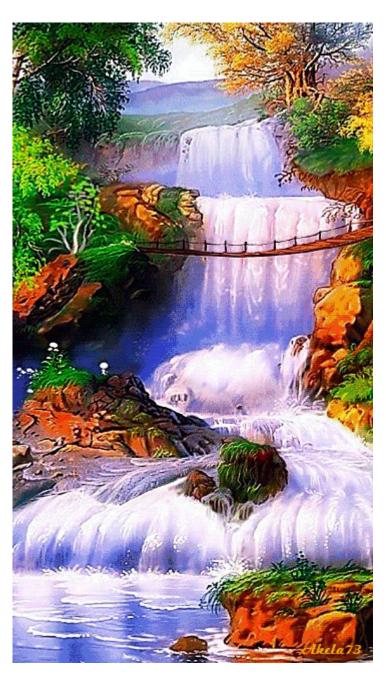
      i+=1
      #将B余下的项添加到C中

      q=B[j]
      C.Add(q)

      j+=1
      return C
```

3. 设计主程序

```
fout=open("abc.out","w+")
                                     #创建第1个多项式顺序表p
p=PolyList()
p.CreateList("abc1.in")
print("第1个多项式:",end=' ',file=fout)
                                     #输出结果写入到abc.out文件中
print(p.getdata(),file=fout)
                                     #第1个多项式顺序表按指数递减排序
p.Sort()
print("排序后结果: ",end=' ',file=fout)
print(p.getdata(),file=fout)
                                     #创建第2个多项式顺序表q
q=PolyList()
q.CreateList("abc2.in")
print("第2个多项式:",end=' ',file=fout)
print(q.getdata(),file=fout)
                                     #第2个多项式顺序表按指数递减排序
q.Sort()
print("排序后结果: ",end=' ',file=fout)
print(q.getdata(),file=fout)
r=p.PolyAdd(q)
                                     #r=p+a
print("相加多项式: ",end=' ',file=fout)
print(r.getdata(),file=fout)
fout.close()
```



欣 赏