

# 数据结构 实验报告



姓名	薛旗	学号	20155362
班级	软信-1503	指导教师	马毅
实验名称	数据结构实验		
开设学期	2016-2017 第一学期		
开设时间	第 1 周——第 10 周		
报告日期	2016.11.28		
评定成绩	评定人		
	评定日期		

东北大学软件学院

## 实验一：顺序表和链表的应用

### 实验目的：

1. 掌握线性表的基本操作（插入、删除、查找）以及线性表合并等运算在顺序存储结构、链式存储结构上的实现。重点掌握链式存储结构实现的各种操作。
2. 掌握线性表的链式存储结构的应用。

### 实验内容：

集合的并、交和差运算

#### [问题描述]

编制一个能演示执行集合的并、交和差运算的程序。

#### [基本要求]

- (1) 集合的元素限定为小写字母 [ ' a' ..... ' z' ]。
- (2) 演示程序以用户和计算机的对话方式实现。

#### [测试数据]

- (1) Set1=" magazine" , Set2=" paper" ,  
Set1 $\cup$ Set2=" aegimnprz" ,  
Set1 $\cap$ Set2=" ae" ,  
Set1-Set2=" gimnz" .
- (2) Set1=" 012oper4a6tion89" , Set2=" error data" ,  
Set1 $\cup$ Set2=" adeinoprt" ,  
Set1 $\cap$ Set2=" aeort" ,  
Set1-Set2=" inp" .

#### [实现提示]

以有序链表来表示集合。

### 实验步骤：

## 1. 实验思路:

首先从键盘读取两组字符分别存入两组数组中, 然后通过后插法建表。调用 `Sort()` 函数将单链表元素排序, 再调用 `Check()` 函数, 删除链表中不是小写字母的元素和重复元素, 得到仅含小写字母的非重复有序元素集合 A 和 B, 再调用函数分别求这两个集合的交并差。

## 2. 函数说明

`main()`: 首先采用尾插法建立单链表, 再调用函数构成有序单链表, 最后用有序单链表表示出两个集合的相关运算。

`InpitElem(char str[], int j)`: 从键盘读取字符存入数组中。

`CreateList_R(LinkList &L, ElemType C[], int n)`: 后插法建表。

`DispList(LinkList L)`: 将内容显示在屏幕上。

`Sort(LinkList &head)`: 单链表元素排序。

`Check(LinkList &head)`: 删除链表中不是小写字母的元素和重复元素。

`Union(LinkList La, LinkList Lb, LinkList &Lc)`: 求两个有序集合的并。两个集合为 A 和 B, 它们的并为集合 C, C 中含有两个集合中的所有元素, 但两个集合中重复的元素只出现一次。

`InterSect(LinkList La, LinkList Lb, LinkList &Lc)`: 求两个有序集合的交。两个集合 A 和 B, 它们的交为集合 C, C 中含有两个集合中重复出现的所有元素。

`Subs(LinkList La, LinkList Lb, LinkList &Lc)`: 求两个有序集合的差。两个集合 A 和 B, 它们的差为集合 C, C 中含有所有属于集合 A 而不属于集合 B 的元素。

## 3. 源代码

```
#include<stdio.h>
#include<malloc.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>
typedef char ElemType;
typedef int Status;
typedef struct LNode//定义单链表节点类型
```

```

{
    ElemType data;//节点的数据域
    struct LNode *next;//节点的指针域
}LNode, *LinkedList;//LinkedList为指向结构体LNode的指针类型

void InputElem(char str[], int j)
{
    fflush(stdin);
    printf("请输入您要运算的第%d组元素: \n", j);
    fflush(stdin);//清空输入缓冲区
    gets(str);
}

void CreateList_R(LinkedList &L, ElemType C[], int n)//后差法建表, 建立带头结点的单链表L
{
    int i;
    LinkedList p, r;
    L = new LNode;
    L->next = NULL;//创建带头节点的空链表
    r = L;
    for (i = 0; i < n; i++)
    {
        p = new LNode;//生成新节点
        p->data = C[i];//将元素值赋给新节点*p的数据域
        p->next = NULL;
        r->next = p;//将新节点*p插入尾节点*r之后
        r = p;//r指向新的尾节点*p
    }
}

void Displist(LinkedList L)
{
    LinkedList p = L->next;
    while (p != NULL)
    {
        printf("%c", p->data);
        p = p->next;
    }
    printf("\n");
}

void Sort(LinkedList &head)//单链表元素排序
{

```

```

LinkedList p = head->next;
LinkedList q, r;
if (p != NULL)//若单链表中有一个或以上的数据节点
{
    r = p->next;//r保存*p节点后继结点的指针
    p->next = NULL;//构造只含一个数据节点的有序表
    p = r;
    while (p != NULL)
    {
        r = p->next;//r保存*p节点后继结点的指针
        q = head;
        while (q->next != NULL && q->next->data < p->data)
            q = q->next;//在有序表中找插入*p的前驱结点*q
        p->next = q->next;//将*p插入到*q之后
        q->next = p;//后移, 比较下一个
        p = r;
    }
}

void Check(LinkedList &head)//删除链表中不是小写字母的元素和重复元素
{
    LinkedList p = head->next;
    LinkedList q;
    while (p->next != NULL)//删除链表中不是小写字母的元素
    {
        if (!(p->data >= 97 && p->data <= 122))
        {
            q = p;
            head->next = p->next;
            p = head->next;
            delete q;
        }
        else
            break;
    }
    p = head->next;
    while (p != NULL && p->next != NULL)//删除链表中重复元素
    {
        while (p->data == p->next->data)
        {
            q = p->next;
            p->next = q->next;
            delete q;
        }
    }
}

```

```

    }
    p = p->next;
}
}

```

`void Union(LinkList La, LinkList Lb, LinkList &Lc)`//求两个有序集合的并

```

{
    LinkList pa = La->next;
    LinkList pb = Lb->next;
    LinkList s, t;
    Lc = new LNode;//创建头结点
    t = Lc;
    while (pa != NULL && pb != NULL)
    {
        if (pa->data < pb->data)
        {
            s = new LNode;//复制节点
            s->data = pa->data;
            t->next = s;
            t = s;
            pa = pa->next;
        }
        else if (pa->data > pb->data)
        {
            s = new LNode;//复制节点
            s->data = pb->data;
            t->next = s;
            t = s;
            pb = pb->next;
        }
        else
        {
            s = new LNode;//复制节点
            s->data = pa->data;
            t->next = s;
            t = s;
            pa = pa->next;//重复的元素只复制一个
            pb = pb->next;
        }
    }
    if (pb != NULL)
        pa = pb;//复制余下的结点
    while (pa != NULL)
    {

```

```

        s = new LNode;//复制节点
        s->data = pa->data;
        t->next = s;
        t = s;
        pa = pa->next;
    }
    t->next = NULL;
}

```

```

void InterSect(LinkList La, LinkList Lb, LinkList &Lc)//求两个有序集合的交
{
    LinkList pa = La->next;
    LinkList pb, s, t;
    Lc = new LNode;
    t = Lc;
    while (pa != NULL)
    {
        pb = Lb->next;
        while (pb != NULL && pb->data < pa->data)
            pb = pb->next;
        if (pb != NULL && pb->data == pa->data)//若B中存在pa节点的值，则复制节点
        {
            s = new LNode;
            s->data = pa->data;
            t->next = s;
            t = s;
        }
        pa = pa->next;
    }
    t->next = NULL;
}

```

```

void Subs(LinkList La, LinkList Lb, LinkList &Lc)
{
    LinkList pa = La->next;
    LinkList pb, s, t;
    Lc = new LNode;
    t = Lc;
    while (pa != NULL)
    {
        pb = Lb->next;
        while (pb != NULL && pb->data < pa->data)
            pb = pb->next;
        if (!(pb != NULL && pb->data == pa->data))//若B中不存在pa节点的值，复制节点

```

```

        {
            s = new LNode;
            s->data = pa->data;
            t->next = s;
            t = s;
        }
        pa = pa->next;
    }
    t->next = NULL;
}

void main()
{
    LinkList La, Lb, Lc;
    ElemType A[50];
    ElemType B[50];
    InputElem(A, 1);
    InputElem(B, 2);
    CreateList_R(La, A, strlen(A));
    CreateList_R(Lb, B, strlen(B));
    Sort(La);
    printf("* 有序集合 A = ");
    DispList(La);
    Sort(Lb);
    printf("* 有序集合 B = ");
    DispList(Lb);
    Check(La);
    printf("* 仅含小写字母的有序集合 A = ");
    DispList(La);
    Check(Lb);
    printf("* 仅含小写字母的有序集合 B = ");
    DispList(Lb);
    Union(La, Lb, Lc);
    printf("* 集合的并 A∪B = ");
    DispList(Lc);
    InterSect(La, Lb, Lc);
    printf("* 集合的交 A∩B = ");
    DispList(Lc);
    Subs(La, Lb, Lc);
    printf("* 集合的差 A-B = ");
    DispList(Lc);
    system("pause");
}

```

**实验结果:**

## 测试一：

```
C:\WINDOWS\system32\cmd.exe
* 开发环境：Microsoft Visual C++ 6.0 *

请输入您要进行运算的第1组元素：
magazine
请输入您要进行运算的第2组元素：
paper
* 有序集合 A = aaegimnz
* 有序集合 B = aeppr
* 仅含小写字母的有序集合 A = aegimnz
* 仅含小写字母的有序集合 B = aepr
* 集合的并  $A \cup B$  = aegimnprz
* 集合的交  $A \cap B$  = ae
* 集合的差  $A - B$  = gimnz
请按任意键继续. . .
```

## 测试二：

```
C:\WINDOWS\system32\cmd.exe
* 开发环境：Microsoft Visual C++ 6.0 *

请输入您要进行运算的第1组元素：
012qper4a6tion89
请输入您要进行运算的第2组元素：
error data
* 有序集合 A = 0124689aeinoopr t
* 有序集合 B = aadeorrrt
* 仅含小写字母的有序集合 A = aeinoprt
* 仅含小写字母的有序集合 B = adeort
* 集合的并  $A \cup B$  = adeinoprt
* 集合的交  $A \cap B$  = aeort
* 集合的差  $A - B$  = inp
请按任意键继续. . .
```

## 实验总结：

本实验的完成基于线性表的链式表示和实现，实验综合考察了单链表的初始化、取值、查找、插入、删除和单链表的创建。本实验采用后插法创建单链表，通过将新节点逐个插入到链表的尾部创建链表。单链表的删除部分，首先要判断要删除元素需要满足的条件，然后再删除。通过本次实验，加深了我对线性表的理解，使我能够运用所学知识解决实际问题，获益匪浅。

## 教师评语或评价表格：（任课教师可根据实际情况，做适当调整）

评语及评价表格的字体颜色为红色

评价表格：

评价内容	具体要求	分值	得分
平时表现	课程设计过程中，无缺勤现象，态度积极，具有严谨的学习态度和认真、踏实、一丝不苟的科学作风。	10	
提交材料	能够按照规范提交课程设计的所有材料（要求在以“学号-姓名”命名的文件夹中，包含实验报告电子版和实验源代码等），材料完备，格式内容等符合要求。	10	
报告质量	实验报告格式规范，符合要求；报告内容充实、正确，实验目的归纳合理到位。	30	
实验内容	能够按实验要求合理设计并开发出程序，功能完整性强，原理及实验结果分析准确，归纳总结充分。	50	
总分			