线性表

| 主讲：戴波 | 电子科技大学计算机学院

目录

CONTENTS

1 线性表的定义

线性表: n个同类型数据元素的有限序列,记为:

$L= (a_1, a_2, \ldots, a_i, \ldots, a_n)$

1 线性表的定义

线性表: n个同类型数据元素的有限序列,记为:

$L=(a_1, a_2, \ldots, a_i, \ldots, a_n)$

L为表名;

i为数据元素a_i 在线性表中的位序;

n为线性表的表长; n=0 时称为空表;

数据元素之间的关系是:

a_{i-1}领先于a_i, a_i领先于a_{i+1}。

称a_{i-1}是a_i的直接前驱, a_{i+1}是a_i的直接后继,

除第一元素a_1外, 均有唯一的前驱;

除最后元素a_n外, 均有唯一的后继;

1 线性表的定义

线性表: n个同类型数据元素的有限序列,记为:

L= (a_1， a_2,…， a_i,…， a_n)

L为表名;

i为数据元素a_i 在线性表中的位序;

n为线性表的表长; n=0 时称为空表;

数据元素之间的关系是:

a_{i-1}领先于a_i， a_i领先于a_{i+1}。

称a_{i-1}是a_i的直接前驱， a_{i+1}是a_i的直接后继，

除第一元素a_1外，均有唯一的前驱;

除最后元素a_n外，均有唯一的后继;

特点:

- a_i的数据类型相同
- 位序i从1开始;
- 前驱与后继

- 练习题

1. 判断题：一个教室里面的5排学生，每排坐6个人，这些人的关系是不是线性表?

2. 电影院售票处排队购票，一共有3个人，如果用ai表示第i个人，则用L=(a1,a2,a3)表示当前排队的人构成的线性表。请问，a2的直接前驱和直接后继分别是谁?

目录

CONTENTS

线性表的抽象数据类型定义

ADT List {

1 数据对象：

$D = \{ a_i \mid a_i \in ElemSet,\ i=1,2,...,n,\ n \geq 0 \}$

2 数据关系：

$R1 = \{ \langle a_{i-1}, a_i \rangle \mid a_{i-1}, a_i \in D,\ i=2,...,n \}$

3 基本操作：

1. 结构初始化操作
2. 结构销毁操作
3. 引用型操作
4. 加工型操作

} ADT List

线性表的抽象数据类型定义

ADT List {

1 数据对象：

$D = \{ a_i \mid a_i \in ElemSet,\ i=1,2,...,n,\ n \geq 0 \}$

2 数据关系：

$R1 = \{ \langle a_{i-1}, a_i \rangle \mid a_{i-1}, a_i \in D,\ i=2,...,n \}$

3 基本操作：

1. 结构初始化操作
2. 结构销毁操作
3. 引用型操作
4. 加工型操作

} ADT List

线性表的基本操作

```
enum Status{
success,fail,fatal,range_error
};
```

```
Status List_Init(SqListPtr L);
void List_Destory(SqListPtr L);
void List_Clear(SqListPtr L);
bool List_Empty(SqListPtr L);
int List_Size(SqListPtr L);
Status List_Retrival(SqListPtr L, int pos, ElemType *elem);
Status List_Locate(SqListPtr L, ElemType elem, int *pos);
Status List_Insert(SqListPtr L, int pos, ElemType elem);
Status List_delete(SqListPtr L, int pos);
Status List_Prior(SqListPtr L, int pos, ElemType * elem);
Status List_Next(SqListPtr L, int pos, ElemType *elem);
```

线性表定义为指针类型SqListPtr，数据元素类型为ElemType

线性表的基本操作—初始化与销毁

```
Status List_Init(SqListPtr L);
void List_Clear(SqListPtr L);
void List_Destory(SqListPtr L);
```

线性表的基本操作—引用型

```
bool List_Empty(SqListPtr L);
int List_Size(SqListPtr L);
Status List_Retrival(SqListPtr L, int pos, ElemType *elem);
Status List_Locate(SqListPtr L, ElemType elem, int *pos);
Status List_Prior(SqListPtr L, int pos, ElemType * elem);
Status List_Next(SqListPtr L, int pos, ElemType *elem);
```

线性表的基本操作—加工型

```
Status List_Insert(SqListPtr L, int pos, ElemType elem);
Status List_delete(SqListPtr L, int pos);
```

线性表的基本操作测试函数

序号	题目	功能描述	测试到的基本操作
1	建立线性表 Test_CreateList	初始化空线性表，通过插入操作逐渐建立具有n个元素的线性表	List_Init; List_Insert; List_Print
2	清除线性表 Test_ClearList	建立非空线性表；打印线性表确定线性表非空；清空线性表；打印线性表确定线性表的所有数据已经被清除	Test_CreateList; List_Print; List_Empty; List_Clear; List_Print
3	定位线性表并查找前驱后继 Test_RetrivalPriorNext	建立线性表；打印线性表；输入要查询的位置并输出该位置的元素；输出该元素的直接前驱与后继	Test_CreateList; List_Print; List_Retirval; List_Prior; List_Next
4	查询 Test_Locate	建立线性表；打印线性表；输出某元素在线性表中的位置	Test_CreateList; List_Print; List_Locate
5	求线性表长度 Test_Length	建立线性表；打印线性表；求线性表的长度	Test_CreateList; List_Print; List_Size

线性表的基本操作测试函数

序号	题目	功能描述	测试到的基本操作
1	建立线性表 Test_CreateList	初始化空线性表，通过插入操作逐渐建立具有n个元素的线性表	List_Init; List_Insert; List_Print
2	清除线性表 Test_ClearList	建立非空线性表；打印线性表确定线性表非空；清空线性表；打印线性表确定线性表的所有数据已经被清除	Test_CreateList; List_Print; List_Empty; List_Clear; List_Print
3	找前驱后继 Test_RetrivalPriorNext	查询的位置并输出该位置的元素；输出该元素的直接前驱与后继	List_Prior; List_Next
4	查询 Test_Locate	建立线性表；打印线性表；输出某元素在线性表中的位置	Test_CreateList; List_Print; List_Locate
5	求线性表长度 Test_Length	建立线性表；打印线性表；求线性表的长度	Test_CreateList; List_Print; List_Size

因为需要查看线性表中的数据，增加打印线性表的基本操作！

讨 论

- 题目：我们要解决问题，总是分析已知条件（输入数据），需要实现的功能（输出），然后寻找输入转换为输出的算法，也就是通过一系列操作步骤对输入数据进行处理，最后转换为输出结果。
- 为什么要研究线性表、树、图这三种数据结构？如何对这些结构进行研究？以线性表为例，怎样的研究是有效的，且能够采用类似方法研究树和图？
- 线性表的这些基本操作有什么用？可以直接拿过来用吗？为什么？和我们待解决的千奇百怪的各种问题有什么联系？