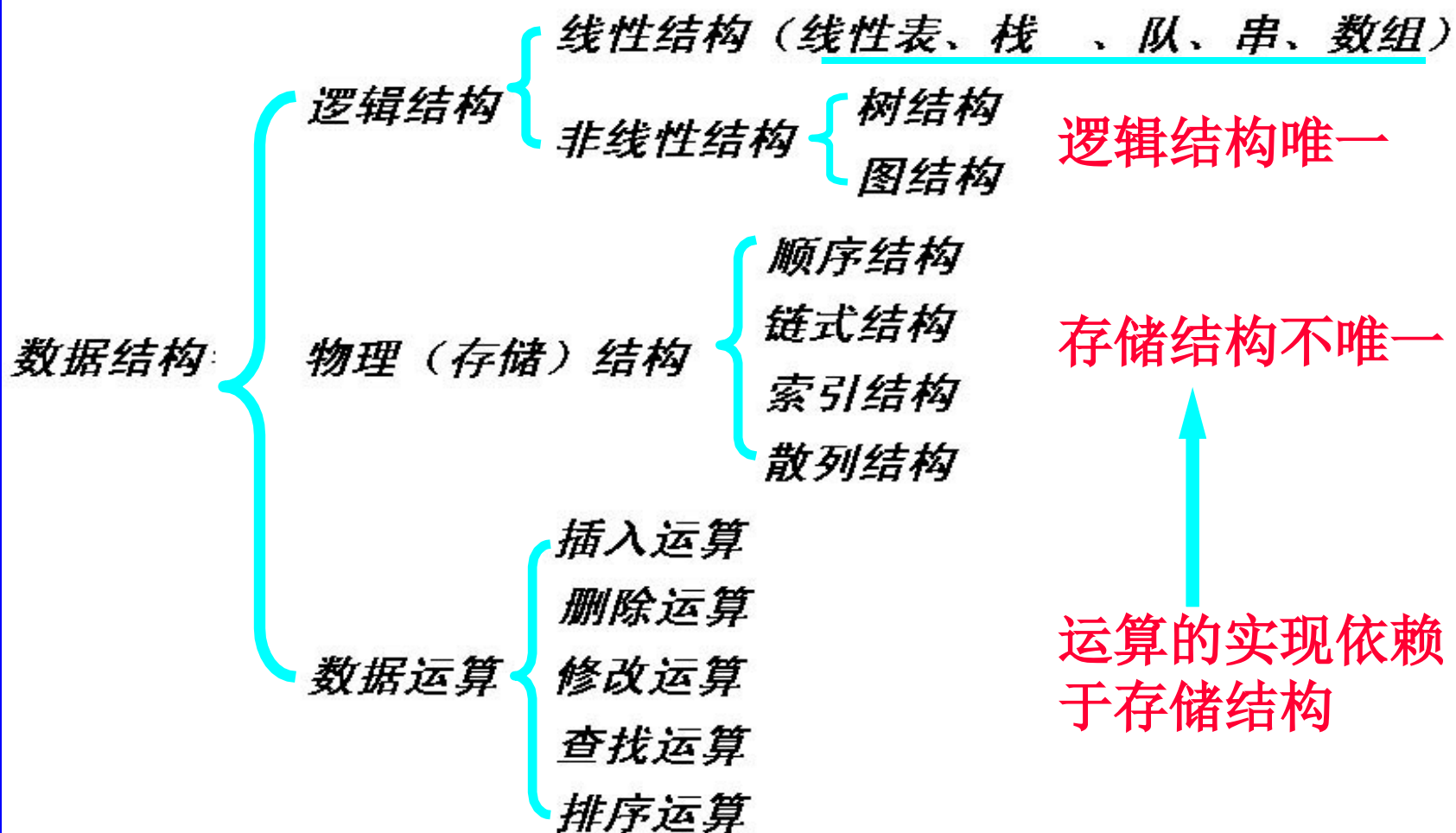


第2章 线性表



数据结构课程的内容

第二章线性表



线性结构的特点:

第二章线性表

1. 集体中必存在唯一的一个**第一个元素**;
2. 集合中必存在唯一的一个**最后一个元素**;
3. 除最后元素在外, 均有**唯一的后继**;
4. 除第一个元素在外, 均有**唯一的前驱**;

线性结构是:

一个数据元素的有序集。

可表示为: (a_1, a_2, \dots, a_n)

2.1 线性表的类型定义

抽象数据类型的定义

第二章 线性表

ADT List{

数据对象:

$$D = \{a_i | a_i \in \text{Elemset}, i=1, 2, \dots, n, n \geq 0\}$$

{n 称为线性表的表长

n=0的线性表称为空表}

数据关系:

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

{设线性表为 $(a_1, a_2, \dots, a_i, \dots, a_n)$

称i为 a_i 在线性表中的位序}

- 基本操作:

- {创建线性表}

- InitList (&L)

- 操作结果: 构造一个空的线性表L

- {销毁线性表}

- DestroyList (&L)

- 初始条件: 线性表L已存在

- 操作结果: 销毁线性表L

{引用型操作}

ListEmpty(L)

ListLenght(L)

PriorElem(L, cur_e, &pre_e)

NextElem(L, cur_e, &next_e)

GetElem(L, i, &e)

LocateElem(L, e, equal())

ListTraverse(L, visit())

ListEmpty(L)

初始条件：线性表L已存在

操作结果：若L为空表，返回TRUE，否则返回FALSE

PriorElem(L, cur_e, &pre_e)

初始条件：线性表L已存在

操作结果：若cur_e是L的元素，且不是第一个，
则用pre_e返回它的前驱，否则操作失败，pre_e无定义

NextElem(L, cur_e, &next_e)

初始条件：线性表L已存在

操作结果：若cur_e是L的元素，且不是最后一个，
则用next_e返回它的后继，否则操作失败，next_e无定义

ListLenght(L)

初始条件：线性表L已存在

操作结果：返回线性表L数据元素个数

GetElem(L, i, &e)

初始条件：线性表L已存在 ($1 \leq i \leq \text{ListLenght}(L)$)

操作结果：用e返回线性表L中第i个数据元素的值

LocatElem(L, e, equal())

初始条件：线性表L已存在，comare()是数据元素判定函数

操作结果：返回线性表L中第1个与e满足关系

comare()的数据元素的位序

ListTraverse(L, visit())

初始条件：线性表L已存在

操作结果：依次对线性表L的每个数据元素调用visit()函数，
一旦visit()失败，则操作失败

{加工型操作}

ClearList(&L)

PutElem(&L, i, &e)

ListInsert(&L, i, e)

ListDelete(&L, i, &e)

第二章 线性表

ClearList(&L)

初始条件: 线性表L已存在

操作结果: 将L重置为空表

PutElem(&L, i, &e)

初始条件: 线性表已存在 ($1 \leq i \leq \text{ListLength}(L) + 1$)

操作结果: L中第i个元素赋值同e的值

第二章 线性表

ListInsert(&L, i, e)

初始条件：线性表已存在 ($1 \leq i \leq \text{ListLength}(L) + 1$)

操作结果：在线性表L中第i个数据元素之前插入新元素e，
L长度加1

ListDelete(&L, i, &e)

初始条件：线性表已存在 ($1 \leq i \leq \text{ListLength}(L)$)

操作结果：删除线性表L中第i个数据元素，用e返回其值，
L长度减1

}ADT List

利用上述线性表的定义，可以完成其他更复杂的操作。

例2-1

假设有两个集合A和B分别用两个线性表LA和LB表示（即线性表中的数据元素即为集合的成员）。

现要求一个新的集合 $A=A \cup B$

此问题可演绎为，要求对线性表作如下操作：

扩大线性表LA，

将存在于线性表LB中而不存在于线性表LA中的数据插入到线性表LA中去。

1, 从线性表LB中依次取得每个数据元素

GetElem(LB, i, &e)

2, 依值在线性表LA中进行查访

LocateElem(LA, e, equal())

3, 若不存在, 则插入之

ListInsert(&LA, n+1, e)


```
void UnionList( List &La, List Lb )
{
    La_len = ListLength( La );
    Lb_len = ListLength( Lb );
    for (i=1; i<=Lb_len; i++){
        GetElem( Lb, i, &e);
        if(!Locate( La, e, equal()))
            ListInsert( &La, ++La_len, e);
    }
}
```

例2-2

已知一个非纯集合B，试构造一个纯集合A，使A中只包含B中所有值各不相同的数据元素。

```
void purse( List &La, List Lb ) {  
    InitList( &La );  
    La_len = ListLength( La );  
    Lb_len = ListLength( Lb );  
    for (i=1; i<=Lb_len; i++) {  
        GetElem( Lb, i, &e);  
        if(!Locate( La, e, equal()))  
            ListInsert( &La, ++La_len, e);  
    }  
}
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/077200151154006056>