**第2章 线性表学习指导**

1、时间

2020年3月24日周二（1-4班3-4节，5-10班、数学、大数据专业5-6节）

2020年3月26日周四（1-4班3-4节，5-10班、数学、大数据专业5-6节）

2020年3月31日周二（1-4班3-4节，5-10班、数学、大数据专业5-6节）

2020年4月02日周四（1-4班3-4节，5-10班、数学、大数据专业5-6节）

4次课共8学时。

2、内容

课前观看视频课件；

腾讯课堂/腾讯会议讲解本章重点和难点；

QQ群交流，中间根据需要采用QQ视频或腾讯视频讲解）；

投票方式答题（点名、检查上课人数）；

问答。

具体安排：

|  |  |
| --- | --- |
| 3月24日  线上 | (1) 线性表的逻辑结构特性和抽象数据类型(ADT)的设计；  (2) 线性表的顺序存储结构和链式存储结构；  (3) 顺序存储实现中的创建、查找、插入和删除等基本操作及相关算法。 |
| 3月26日  线上 | (4) 链式存储实现中单链表的创建、查找、插入和删除等基本操作及相关算法；  (5) 双向链表的插入和删除等基本操作及相关算法；  (6) 循环链表的特点及创建、查找、插入和删除等基本操作及相关算法。 |
| 3月31日  线上 | (7) 栈与队列的定义、特点和性质；  (8) 栈与队列的设计、实现以及基本操作和相关算法；  (9) 栈和队列在表达式求值、括号匹配、数制转换问题中的应用。 |
| 4月2日  （H308？） | (10) 串的定义、性质和特点；  (11) 串的设计、实现方法和基本操作；  (12) 串的简单模式匹配算法及KMP算法等。 |
| (13) 数组的存储表示方法；  (14) 数组在存储结构中的地址计算方法；  (15) 特殊矩阵的压缩存储方法；  (16) 稀疏矩阵的两种压缩存储方法；  (17) 以三元组表示稀疏矩阵时矩阵运算所采用的算法及实现；  (18) 广义表的定义、特点和性质。 |

3、本章知识要点

**数据对象中除第一个元素以外的任何一个元素有唯一的前驱，除最后一个元素以外的每一个元素有唯一的后继元素。**

**3.1 线性表逻辑结构**

**线性表**是n(n>=0)个数据元素的有限序列，同一线性表中的数据元素必定具有相同特性，即属于同一数据对象，相邻数据元素之间存在序偶<a*i*,a*i-1*>关系。n定义为线性表的长度；n为0表示该线性表为空表。常把线性表表示为：

（a*1*, a*2*, a*3*, … , a*i-1*, a*i*, a*i+1*, … , a*n*）

简言之，在线性表中，除第一个元素无前驱、最后一个元素无后继之外，其他任何一个元素都有唯一的直接前驱元素和唯一的一个直接后继元素。

序偶<a*i*,a*i+1*>,*i*=1….n-1,称a*i*为a*i+1*的直接前驱，a*i+1*为a*i*的直接后继。

序：指元素之间的“位序”，而非元素值的升序、降序，或其他顺序。

**3.2、线性表的ADT操作及实现**

**数学模型：**

线性表 LIST = ( D , R)

D = { a*i* | a*i* ∈Elementset , *i* = 1, 2, …, n, n ≥ 0 }

R = { H }

H = { <a*i-1*, a*i*> | a*i-1*, a*i* ∈ D , *i* = 2 , … , n }

**操作：**

1. Initlist（L）
2. Listlength（L）
3. GetNode（L，i）
4. LocateNode（L，x）
5. InsertList（L，x，i）
6. Delete（L，i）

**注意与教材的区别：**

1. ListInsert(L, p, e)；
2. LocateElem(L,e, Compare())；
3. GetElem( L,p,&e)；
4. ListDelete(&L,p,&e)；
5. PriorElem( L,cur\_e, &pre.e)
6. NextElem( L,cur\_e,&next\_e)；
7. ClearList(&L)；
8. ListrFirst( L) ；
9. ListEnd(L)；, …..

**3.3 线性表存储结构**

**（1）顺序存储**

线性表中任一数据元素的存储位置为：LOC(a*i*)=LOC(a*1*)+(*i-1*)\*L

一维数组实现线性表，也就是采用一位数组作为线性表的存储结构，利用一维素组元素的地址连续来表示线性表元素的位序，这是一种**静态存储结构**。

例如：ElementType A[Max];

思考：int A[Max] , \*pa;

前提：pa = A ; 或 pa = &A[0];

请问：A、pa、A[i]、\*(pa+i) 、pa[i]、\*(A+i) 之间的关系？i=1…Max-1

操作：

* 插入：平均移动结点次数为*n/2*；平均时间复杂度均为O(*n*)。
* 删除：平均移动结点次数为(*n-1*)/*2*；平均时间复杂度均为O(*n*)。

**（2）链式存储**

线性链表是一种**动态存储结构**，所占用的存储空间是在程序的执行过程中得到的，当线性链表要增加一个结点时，向系统申请一个存储空间，删除结点时要将空间释放。

由线性链表的结点定义，每个结点中均只含有一个指针域，用于指向其后继结点，故也称单链表。

循环链表是线性表的另一种形式的链式存储表示。它的特点是表中最后一个结点的指针域指向头结点，整个链表成为一个由链指针相链接的环，并且可将头指针设成指向最后一个结点（尾指针）。空的循环链表由只含一个自成循环的头结点表示。

若每个结点中均含有两个指针域，一个用于指向其后继结点，另一个用于指向其前驱结点，则称其为双向链表。

若双向链表中的两个链均构成回路，则称为双向循环链表。

**重点理解：线性链表头结点的作用。**

操作：

* 插入：平均移动结点次数为?；平均时间复杂度均为O(?)。
* 删除：平均移动结点次数为?；平均时间复杂度均为O(?)。

**（3）线性链表静态存储结构与动态存储结构的比较**

**（4）风别针对每一种存储结构，实现每一个操作**

**3.4栈和队列**

**栈**是限定只能在表的一端（表尾）进行插入和删除操作的线性表；允许插入和删除的一端，称为栈顶(*top*)；另一端则称为栈底(*bottom*)；栈又叫做后进先出（LIFO）的线性表。

为了指示当前的栈顶元素，需设一个指针*top*指示当前栈顶的位置，称为**栈顶指针**。

**队列**也是受限的线性表。限定只能在队列的一端插入元素，另一端删除元素。插入元素的一端是队尾(*rear*)，删除元素的一端是队头(*front*)。队列具有先进先出的特点，因此称为先进先出（FIFO）线性表。

**重点**：（1）栈和队列的存储结构;

（2）栈和队列的ADT操作;

（3）栈和队列的应用。

**3.5串**

**串**是由*n*(*n*>=0)个任意字符组成的有限序列。当*n*为零时，称为空串。由一个或多个空格符组成的串称为空格串。

**子串**：串中任意连续的字符组成的子序列称为该串的子串。

**主串**：包含子串的串。

**子串的位置**：子串的第一个字符在主串中的序号称为子串的位置。

**两个串相等**：当且仅当两个串的长度相等且对应位置上的字符都相同。

**重点：**（1）串的存储结构及比较；

（2）串的操作；

**（3）模式匹配算法（KMP）。**

**3.6 数组与广义表**

**任何数组都是一个地址连续的一维向量。一维数组是；二维数组也是，所不同的是二维数组的每一个元素又是一个一位数组；……**

**数组**是连续的、有限的、有序的、同构的数据元素的集合；LOC(a*i*)=LOC(a*1*)+(i*-1*)\*L（一维数组）

LOC(a*ij*)=LOC(a*11*)+[(*i-1*)\*N+*j-1*]\*L（行序为主的存储方式）

**特殊矩阵**：三角矩阵、对称矩阵或对角矩阵，值相同或零元素的分布在矩阵中有一定的规律。

**稀疏矩阵**：非零元素个数远小于矩阵（零）元素个数。

**压缩存储**：为多个值相同的元只分配一个存储空间以节省存储空间；对零元不分配空间。

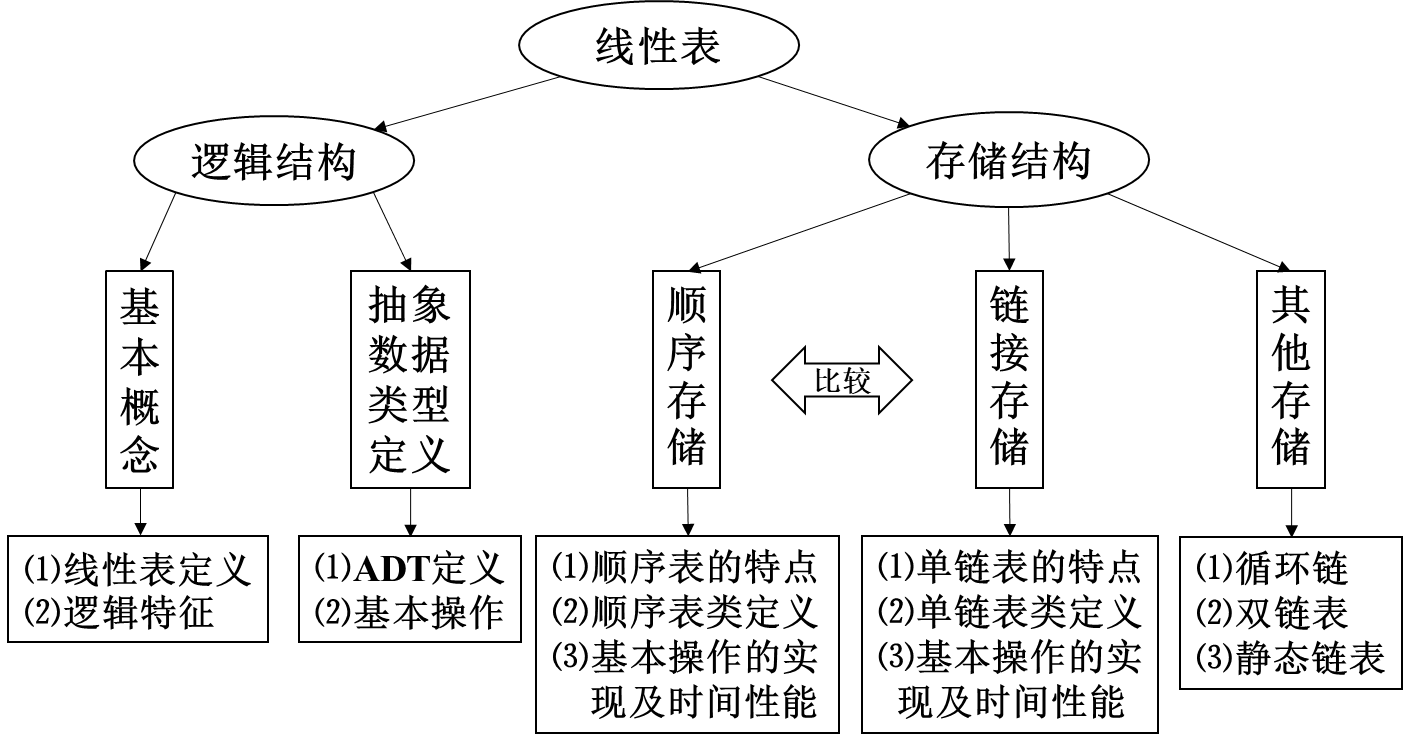
**重点：**矩阵压缩存储后的转置运算的两种实现方法及其时间复杂度的比较。

**广义表**：LS=(a*1*,a*2*,…,a*n*)LS为广义表的名称，*n*为广义表的长度，a*i*可以是单个元素，也可以是广义表，称为LS的原子和子表。当广义表非空时，称第一个元素a*1*为表头（*Head*），称其余元素组成的表（a*2*,a*3*,…,a*n*）为表尾（*Tail）*。

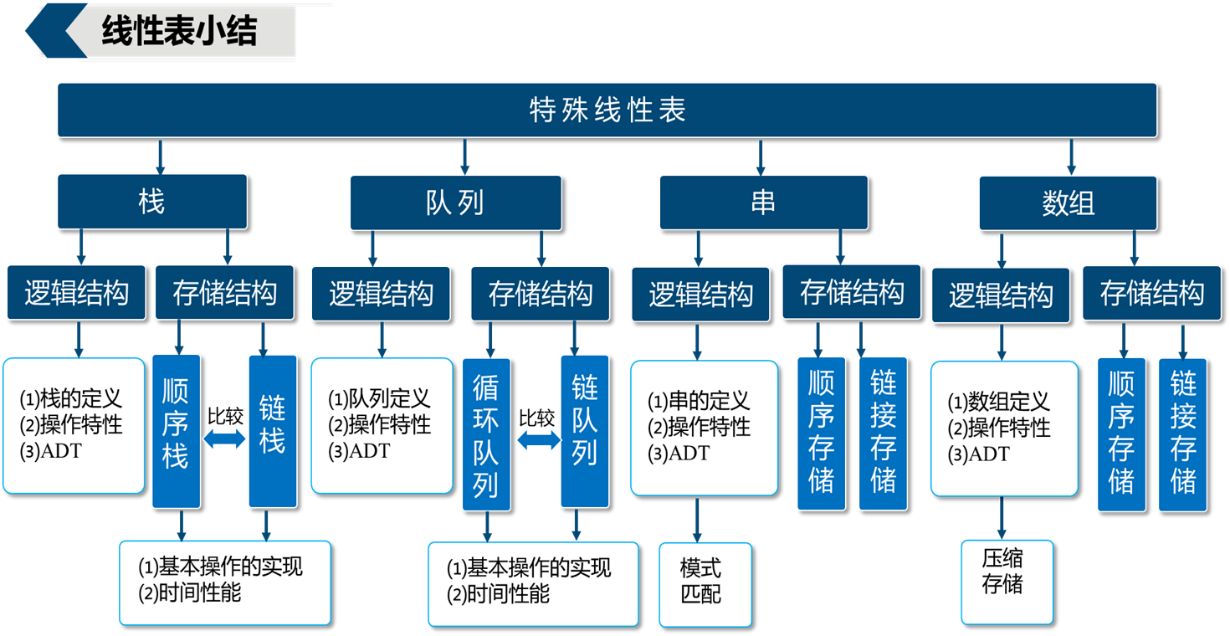
要求理解广义表的定义，能读懂广义表的存储结构。

**3.7 本章小结**

**1）知识点**

****

**2）小结**



**4、习题**

**（1）单项选择题**

1） 一个向量（即一批地址连续的存储单元）第一个元素的存储地址是100，

每个元素的长度为2，则第5个元素的地址是( )。

A. 110 B. 108 C. 100 D. 120

2）线性表的顺序存储结构是一种( )的存储结构，而链式存储结构是一种

( )的存储结构。

A．随机存取 B．索引存取 C．顺序存取 D．散列存取

3）线性表的逻辑顺序与存储顺序总是一致的，这种说法( )。

A. 正确 B. 不正确

4）线性表若采用链式存储结构时，要求内存中可用存储单元的地址( )。

A. 必须是连续的 B. 部分地址必须是连续的

C. 一定是不连续的 D. 连续或不连续都可以

5）在以下的叙述中，正确的是( )。

A. 线性表的顺序存储结构优于链表存储结构

B. 线性表的顺序存储结构适用于频繁插入/删除数据元素的情况

C. 线性表的链表存储结构适用于频繁插入/删除数据元素的情况

D. 线性表的链表存储结构优于顺序存储结构

6）每种数据结构都具备三个基本运算：插入、删除和查找，这种说法( )。

A. 正确 B. 不正确

7）不带头结点的单链表head为空的判定条件是( )。

A. head= =NULL B. head->next= =NULL

C. head->next= =head D. head!=NULL

8）带头结点的单链表head为空的判定条件是( )。

A. head= =NULL B. head->next= =NULL

C. head->next= =head D. head!=NULL

9）非空的循环单链表head的尾结点（由p所指向）满足( )。

A. p->next= =NULL B. p= =NULL

C. p->next= =head D. p= =head

10）在双向循环链表的p所指结点之后插入s所指结点的操作是( )。

A. p->right=s; s->left=p; p->right->left=s; s->right=p->right;

B. p->right=s; p->right->left=s; s->left=p; s->right=p->right;

C. s->left=p; s->right=p->right; p->right=s; p->right->left=s;

D. s->left=p; s->right=p->right; p->right->left=s; p->right=s;

11）在一个单链表中，已知q所指结点是p所指结点的前驱结点，若在q和p

之间插入s结点，则执行( )。

A. s->next=p->next; p->next=s; B. p->next=s->next; s->next=p;

B. q->next=s; s->next=p; C. p->next=s; s->next=q;

12）在一个单链表中，若p所指结点不是最后结点，在p之后插入s所指结

点，则执行( )。

A. s->next=p; p->next=s; B. s->next=p->next; p->next=s;

C. s->next=p->next; p=s; C. p->next=s; s->next=p;

13）在一个单链表中，若删除p所指结点的后续结点，则执行( )。

A. p->next= p->next->next； B. p= p->next; p->next= p->next->next；

C. p->next= p->next; D. p= p->next->next；

14）从一个具有n个结点的单链表中查找其值等于x结点时，在查找成功的情

况下，需平均比较( )个结点。

A. n B. n/2 C. (n-1)/2 D. (n+1)/2

15）在一个具有n个结点的有序单链表中插入一个新结点并仍然有序的时间复

杂度是( )。

A. O(1) B. O(n) C. O (n2) D. O (nlog2n)

16）给定有n个元素的向量，建立一个有序单链表的时间复杂度是( )。

A. O(1)） B. O(n) C. O (n2) D. O (n\*log2n)

**（2）填空题（将正确的答案填在相应的空中）**

1）单链表可以做\_\_\_\_\_\_\_的链接存储表示。

2） 在双链表中，每个结点有两个指针域，一个指向\_\_\_\_\_\_\_，另一个指向\_\_\_\_\_\_。

3）在一个单链表中p所指结点之前插入一个s (值为e)所指结点时，可执行如

下操作：

q=head;

while (q->next!=p) q=q->next;

s= **new** Node; s->data=e;

q->next= ; //填空

s->next= ; //填空

4）在一个单链表中删除p所指结点的后继结点时，应执行以下操作：

q= p->next;

p->next=\_\_\_\_\_\_\_; //填空

**delete** \_\_\_\_\_\_\_; //填空

5）在一个单链表中p所指结点之后插入一个s所指结点时，应执行

s->next=\_\_\_\_\_\_\_和p->next=\_\_\_\_\_\_\_的操作。

6）对于一个具有n个结点的单链表，在已知p所指结点后插入一个新结点的

时间复杂度是\_\_\_\_\_\_\_；在给定值为x的结点后插入一个新结点的时间复杂

度是\_\_\_\_\_\_\_。

**（3）算法设计题**

1. 设m×n阶稀疏矩阵A有t个非零元素，其三元组表表示为LTMA[1:(t+1)，1:3]，试问:非零元素的个数t达到什么时候时 用LTMA表示A才有意义?
2. 设顺序表va中的数据元数递增有序。试写一算法，将*x*插入到顺序表的适当位置上，以保持该表的有序性。
3. 试写一算法，实现顺序表的就地逆置，即利用原表的存储空间将线性表（a*1*, a*2*,…. a*n*）逆置为(a*n*, a*n-1*,…., a*1*)。
4. 已知由**n**个整数构成序列的数据分布为先下降再上升，即一开始数据是严格递减的，后来数据是严格递增的，设计尽可能高效算法，找到序列中的最小值。
5. 已知由n个整数构成序列的数据分布为先上升再下降， 即一开始数据是严格递增的，后来数据是严格递减的，设计尽可能高效算法，找到序列中的最大值。
6. 输入一个已经按升序排序过的数组和一个数字，在数组中查找两个数，使得它们的和正好是输入的那个数字。
7. 给定含有n个元素的两个有序(非降序)整型数组a和b。合并两个数组中的元素到整型数组c，要求去除重复元素并保持c有序(非降序) 。如：
8. 给定两个有序整型数组a和b，各有n个元素，求两个数组中满足给定和的数对，即对a中元素i和b中元素j，满足i + j = d(d已知)。
9. ⑵ 已知数组A[*n*]中的元素为整型，设计算法将其调整为左右两部分，左边所有元素为奇数，右边所有元素为偶数，并要求算法的时间复杂度为Ｏ(n)。
10. 在值为x的结点后插入一个新的结点y, 若x不存在，则插入在表尾。
11. 已知线性表中的元素以值递增有序排列，并以单链表作存储结构。试写一算法，删除表中所有大于x且小于y的元素（若表中存在这样的元素）同时释放被删除结点空间。
12. 试写一算法，实现单链表的就地逆置(要求在原链表上进行)。
13. 已知链表的头结点, 逆序打印这个链表, 要求时间复杂度为O(n)。
14. 如何判断单链表中是否存在环?若存在环，设计算法找到环的入口点。
15. 线性链表交叉问题，设计算法判断两个单链表是否相交。