目录

[**1、** **数据结构** 2](#_Toc497163036)

[**1.1** **线性表** 2](#_Toc497163037)

[**定义** 2](#_Toc497163038)

[**特征** 2](#_Toc497163039)

[**基本操作** 2](#_Toc497163040)

[**存储结构** 2](#_Toc497163041)

[**结构特点** 3](#_Toc497163042)

[**线性表的推广** 3](#_Toc497163043)

[1.2 3](#_Toc497163044)

1. **数据结构**
   1. **线性表**

**定义**

线性表（linear list）是数据结构的一种，一个线性表是n个具有相同特性的数据元素的有限序列。数据元素是一个抽象的符号，其具体含义在不同的情况下一般不同。

在稍复杂的线性表中，一个数据元素可由多个数据项（item）组成，此种情况下常把数据元素称为记录（record），含有大量记录的线性表又称文件（file）。

线性表中的个数n定义为线性表的长度，n=0时称为空表。在非空表中每个数据元素都有一个确定的位置，如用ai表示数据元素，则i称为数据元素ai在线性表中的位序。

线性表的相邻元素之间存在着序偶关系。如用（a1，…，ai-1，ai，ai+1，…，an）表示一个顺序表，则表中ai-1领先于ai，ai领先于ai+1，称ai-1是ai的直接前驱元素，ai+1是ai的直接后继元素。当i=1,2，…，n-1时，ai有且仅有一个直接后继，当i=2，3，…，n时，ai有且仅有一个直接前驱

**特征**

集合中必存在唯一的一个“第一元素”。

集合中必存在唯一的一个 “最后元素” 。

除最后一个元素之外，均有 唯一的后继(后件)。

除第一个元素之外，均有 唯一的前驱(前件)。

**基本操作**

1）MakeEmpty(L) 这是一个将L变为空表的方法

2）Length（L） 返回表L的长度，即表中元素个数

3）Get（L，i） 这是一个函数，函数值为L中位置i处的元素（1≤i≤n）

4）Prior（L，i） 取i的前驱元素

5）Next（L，i） 取i的后继元素

6）Locate（L，x） 这是一个函数，函数值为元素x在L中的位置

7）Insert（L，i，x）在表L的位置i处插入元素x，将原占据位置i的元素及后面的元素都向后推一个位置

8）Delete（L，p） 从表L中删除位置p处的元素

9）IsEmpty(L) 如果表L为空表(长度为0)则返回true，否则返回false

10）Clear（L）清除所有元素

11）Init（L）同第一个，初始化线性表为空

12）Traverse（L）遍历输出所有元素

13）Find（L，x）查找并返回元素

14）Update（L，x）修改元素

15）Sort（L）对所有元素重新按给定的条件排序

16) strstr(string1,string2)用于字符数组的求string1中出现string2的首地址

**存储结构**

线性表主要由**顺序表**示或**链式表**示。在实际应用中，常以栈、队列、字符串等特殊形式使用。

**顺序表**示指的是用**一组地址连续的存储单元依次存储线性表的数据元素**，称为线性表的**顺序存储结构**或**顺序映像**（sequential mapping）。它以“物理位置相邻”来表示线性表中数据元素间的逻辑关系，可随机存取表中任一元素。

优点：具有简单、运算方便等优点，特别是对于小线性表或长度固定的线性表，采用顺序存储结构的优越性更为突出

缺点：

1.顺序存储插入与删除一个元素，必须移动大了的数据元素，以此对大的线性表，特别是在元素的插入和删除很频繁的情况下，采取顺序存储很是不方便，效率低；

2.顺序存储空间容易满，出现上溢，程序访问容易出问题，顺序存储结构下，存储空间不便扩充；

3.顺序存储空间的分配问题，分多了浪费，分少了空间不足上溢

对于大的线性表，特别是元素变动频繁的大线性表不宜采用顺序存储空间，而采用链式存储结构

**链式表**示指的是用**一组任意的存储单元**存储线性表中的数据元素，称为线性表的**链式存储结构**。它的存储单元可以是连续的，也可以是不连续的。在表示数据元素之间的逻辑关系时，除了存储其本身的信息之外，还需存储一个指示其直接后继的信息（即直接后继的存储位置），这两部分信息组成数据元素的存储映像，称为结点（node）。它包括两个域；存储数据元素信息的域称为数据域；存储直接后继存储位置的域称为指针域。指针域中存储的信息称为指针或链

单向链表、双向链表、循环链表

**结构特点**

1.均匀性：虽然不同数据表的数据元素可以是各种各样的，但对于同一线性表的各数据元素必定具有相同的数据类型和长度。

2.有序性：各数据元素在线性表中的位置只取决于它们的序号，数据元素之前的相对位置是线性的，即存在唯一的“第一个“和“最后一个”的数据元素，除了第一个和最后一个外，其它元素前面均只有一个数据元素(直接前驱)和后面均只有一个数据元素（直接后继）。

**线性表的推广**

时间有序表、排序表、和频率有序表都可以看做是线性表的推广。如果按照结点到达结构的时间先后，作为确定结点之间关系的，这样一种线性结构称之为时间有序表。例如，在红灯前停下的一长串汽车，最先到达的为首结点，最后到达的为尾结点；在离开时最先到达的汽车将最先离开，最后到达的将最后离开。这些汽车构成理一个队列，实际上就是一个时间有序表。栈和队列都是时间有序表。频率有序表是按照结点的使用频率确定它们之间的相互关系的，而排序表是根据结点的关键字值来加以确定的。

* 1. **栈**

**定义**

栈是限定仅在表头进行插入和删除操作的线性表

栈作为一种数据结构，是一种只能在一端进行插入和删除操作的特殊线性表。它按照先进后出的原则存储数据，先进入的数据被压入栈底，最后的数据在栈顶，需要读数据的时候从栈顶开始弹出数据（最后一个数据被第一个读出来）。栈具有记忆作用，对栈的插入与删除操作中，不需要改变栈底指针。

栈是允许在同一端进行插入和删除操作的特殊线性表。允许进行插入和删除操作的一端称为栈顶(top)，另一端为栈底(bottom)；栈底固定，而栈顶浮动；栈中元素个数为零时称为空栈。插入一般称为进栈（PUSH），删除则称为退栈（POP）。栈也称为后进先出表。

栈可以用来在函数调用的时候存储断点，做递归时要用到栈！

在计算机系统中，栈则是一个具有以上属性的动态内存区域。程序可以将数据压入栈中，也可以将数据从栈顶弹出。在i386机器中，栈顶由称为esp的寄存器进行定位。压栈的操作使得栈顶的地址减小，弹出的操作使得栈顶的地址增大。

栈在程序的运行中有着举足轻重的作用。最重要的是栈保存了一个函数调用时所需要的维护信息，这常常称之为堆栈帧或者活动记录。堆栈帧一般包含如下几方面的信息：

1．函数的返回地址和参数

2． 临时变量：包括函数的非静态局部变量以及编译器自动生成的其他临时变量。

**算法**

1．进栈（PUSH）算法

①若TOP≥n时，则给出溢出信息，作出错处理（进栈前首先检查栈是否已满，满则溢出；不满则作②）；

②置TOP=TOP+1（栈指针加1，指向进栈地址）；

③S(TOP)=X，结束（X为新进栈的元素）；

2．退栈（POP）算法

①若TOP≤0，则给出下溢信息，作出错处理(退栈前先检查是否已为空栈， 空则下溢；不空则作②)；

②X=S(TOP)，（退栈后的元素赋给X）：

③TOP=TOP-1，结束（栈指针减1，指向栈顶）。

**实现**