



线性表是算法题命题的重点。这类算法题的实现比较容易且代码量较少，但是要求具有最优的性能（时间/空间复杂度），才能获得满分。因此，应牢固掌握线性表的各种基本操作（基于两种存储结构），在平时的学习中多注重培养动手能力。另需提醒的是，算法最重要的是思想！考场上的时间紧迫，在试卷上不一定要求代码具有实际的可执行性，因此应尽力表达出算法的思想和步骤，而不必过于拘泥所有细节。此外，采用时间/空间复杂度较差的方法也能拿到大部分分数，因此在时间紧迫的情况下，建议直接采用暴力法。注意，算法题只能用C/C++语言实现。

# 线性表的定义和基本操作

## 线性表的定义

线性表是具有相同的数据类型的 $n(n \geq 0)$ 个数据元素的有限序列，其中 $n$ 为表长，当 $n=0$ 时线性表是一个空表。若用 $L$ 命名线性表，则其一般表示为

$$L = (a_1, a_2, \dots, a_i, a_{i+1}, \dots, a_n)$$

式中， $a_1$ 是唯一的“第一个”数据元素，又称表头元素； $a_n$ 是唯一的“最后一个”数据元素，又称表尾元素。除第一个元素外，每个元素有且仅有一个直接前驱。除最后一个元素外，每个元素有且仅有一个直接后继（“直接前驱”和“前驱”、“直接后继”和“后继”通常被视为同义词）。以上就是线性表的逻辑特性，这种线性有序的逻辑结构正是线性表名字的由来。

由此，我们得出线性表的特点如下：

- 表中元素的个数有限。
- 表中元素具有逻辑上的顺序性，表中元素有其先后次序。
- 表中元素都是数据元素，每个元素都是单个元素。
- 表中元素的数据类型都相同，这意味着每个元素占有相同大小的存储空间。
- 表中元素具有抽象性，即仅讨论元素间的逻辑关系，而不考虑元素究竟表示什么内容。

线性表是一种逻辑结构，表示元素之间一对一的相邻关系。顺序表和链表是指存储结构，两者属于不同层面的概念，因此不要将其混淆。

## 线性表的基本操作

一个数据结构的基本操作是指其最核心、最基本的操作。其他复杂的操作可通过调用其基本操作来实现。线性表的主要操作如下。

InitList(&L)	初始化表。构造一个空的线性表。
Length(L)	求表长。返回线性表L的长度，即L中数据元素的个数。
LocateElem(L,e)	按值查找操作。在表L中查找具有给定关键字值的元素。

GetElem(L,i)	按位查找操作。获取表L中第i个位置的元素的值。
ListInsert(&L,i,e)	插入操作。在表L中的第i个位置上插入指定元素e。
ListDelete(&L,i,&e)	删除操作。删除表L中第i个位置的元素，并用e返回删除元素的值。
PrintList(L)	输出操作。按前后顺序输出线性表L的所有元素值。
Empty(L)	判空操作。若L为空表，则返回true，否则返回false。
DestroyList(&L)	销毁操作。销毁线性表，并释放线性表L所占用的内存空间。