物理结构是逻辑结构在计算机中的存储形式

数据的存储结构中的顺序存储结构能正确反映数据元素之间的逻辑关系，而链式存储结构不能

线性表

（1、线性表是具有（ 零个 ）或 （多个 ）数据元素的（ 有限 ）序列。

线性表中数据元素之间的关系是（ 一对一 ）的关系，即除了第一个和最后一个数据元素之外，其他数据元素都是（ 首尾相连 ）的

（2、线性表的基本特征：

1、第一个数据元素没有前驱元素

2、最后一个数据元素没有后继元素

3、其余每个数据元素只有一个前驱元素和一个后继元素

4、抽象数据类型

（3、线性表一般包括（ 插入 ）、（ 删除 ）、（ 查找 ）等基本操作

（4、线性表按物理存储结构的不同可分为 顺序表（顺序存储） 和 链表（链式存储）

1、顺序表（存储结构连续）

2、链表（存储结构上不连续，逻辑上连续）

顺序表：

顺序表是在计算机内存中以（ 一组地址连续 ）的存储单元依次存储数据元素的线性结构。

顺序表效率分析：

顺序表插入和删除一个元素，最好情况下其时间复杂度（这个元素在最后一个位置）为O(1),最坏情况下其时间复杂度为O(n)

顺序表支持随机访问，读取一个元素的时间复杂度O(1)

顺序表的优缺点：

优点：支持随机访问

缺点：插入和删除操作需要移动大量的元素，造成存储空间的碎片。

顺序表适合元素个数变化不大，且更多是读取数据的场合。

Python中的 list 和 tuple 两种类型采用了顺序表的实现技术，具有前面讨论的顺序表的所有性质。

链表：

为什么需要链表：

（1、顺序表的构建需要预先知道数据大小来申请连续的存储空间，而在进行扩充时又需要进行数据的搬迁，所以使用起来并不是很灵活。

（2、链表结构可以充分利用计算机内存空间，实现灵活的内存动态管理。

链表的定义：

链表是一种常见的基础数据结构，是一种线性表，但是不像顺序表一样连续存储数据，而是在每一个节点（数据存储单元）里存放下一个节点的位置信息（即地址）

顺序表存储数据信息 而链表存储下一个节点的位置（ 地址 ）

链表的分类：

1、 单向链表

2、单向循环链表

3、双向链表

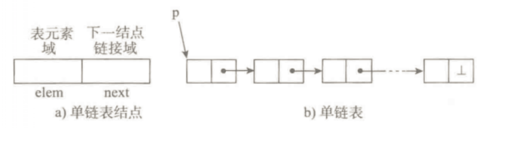
单向链表：

单向链表也叫单链表，是链表中最简单的一种形式，它的每个节点包含两个域，一个信息域（ 元素域，也就是存储的元素 ）和一个链接域（存储的是节点，也就是地址）。这个链接指向链表中的下一个节点，而最后一个节点的链接域则指向一个空值。

表元素域elem用来存放具体的数据。

链接域next用来存放下一个节点的位置

变量p指向链表的头节点（首节点）的位置，从p出发能找到表中的任意节点



节点实现：

Class SingleNode(object):

‘’’单链表的节点’’’

def \_\_init\_\_(self,item):

#item存放数据元素

Self.item=item

#next是下一个节点的标识

Self.next=None

单链表的典型操作

Is\_empty() 链表是否为空

Length()链表长度

Travel()遍历整个链表

Add(item)链表头部添加元素

Append(item)链表尾部添加元素

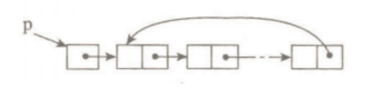
Insert(pos,item)指定位置添加元素

Remove(item)删除节点

Search(item)查找结点是否存在

单向循环链表

单链表的一个变形时单向循环链表，链表中最后一个节点的next域不再为None,而是指向链表的头节点。



双向链表

一种更复杂的链表是“双向链表”或“双面链表”。

每个节点有两个链接：

1. 一个指向前一个节点，当此节点为第一个节点时，指向空值；
2. 而另一个指向下一个节点，当此节点为最后一个节点时，指向空值

