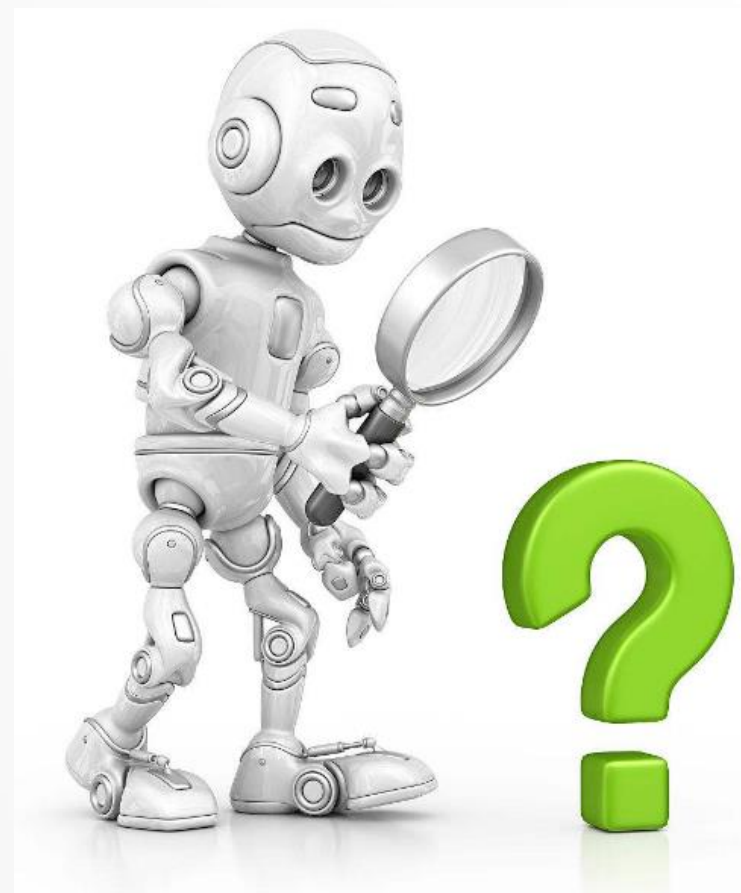


# 线性表的存储结构 | 链式存储结构

- 顺序存储结构可以存储线性表的数据，支撑线性表的基本和复杂操作的实现

## 问题：

- 可是我们无法知道到底预先分配多大空间合适！
- 插入删除开销很大



## 解决方案

- 一个数据一个空间，数据之间不连续，避免一次性大空间分配失败，或者使用过程中空间不足的问题



地址不连续

- 一个数据分配一个空间，数据之间不连续，如何表示元素之间的关系？



每个空间还需要  
存储元素关系

# 线性表的链式存储 | 单链表

## 一、单链表

用一组地址任意的存储单元存放线性表中的数据元素。

数据域 (数据元素)

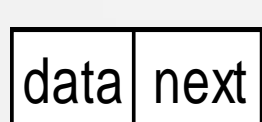
+ 指针域 (指示后继元素存储位置)

= 结点

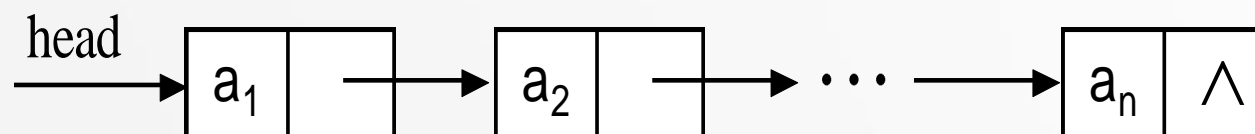
以“结点的序列”表示线性表 —— 称作链表

# 一、单链表

- 逻辑结构
  - $(a_1, a_2, \dots, a_{i-1}, \mathbf{a_i}, a_{i+1}, \dots, a_n)$
- 物理结构



(a) 结点结构

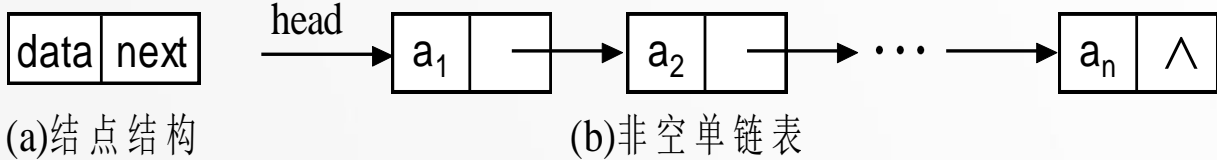


(b) 非空单链表

# 一、单链表

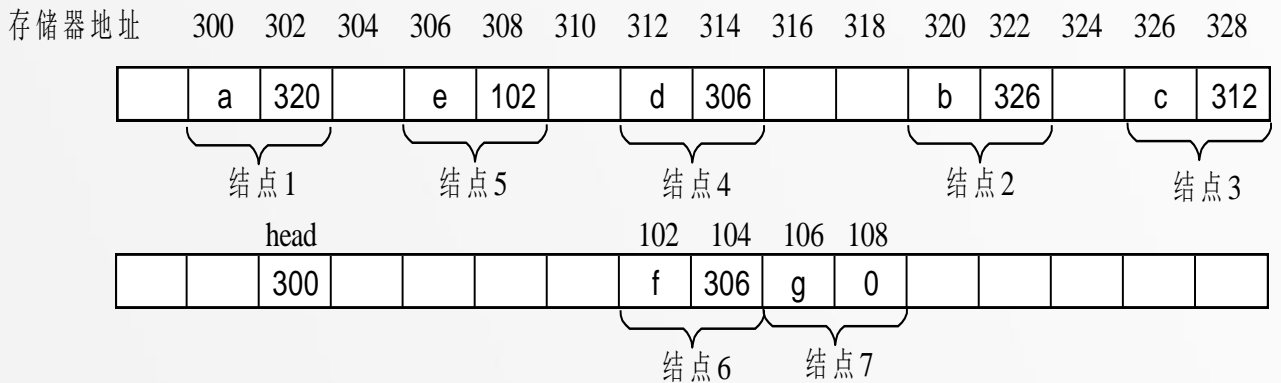
- 逻辑结构

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

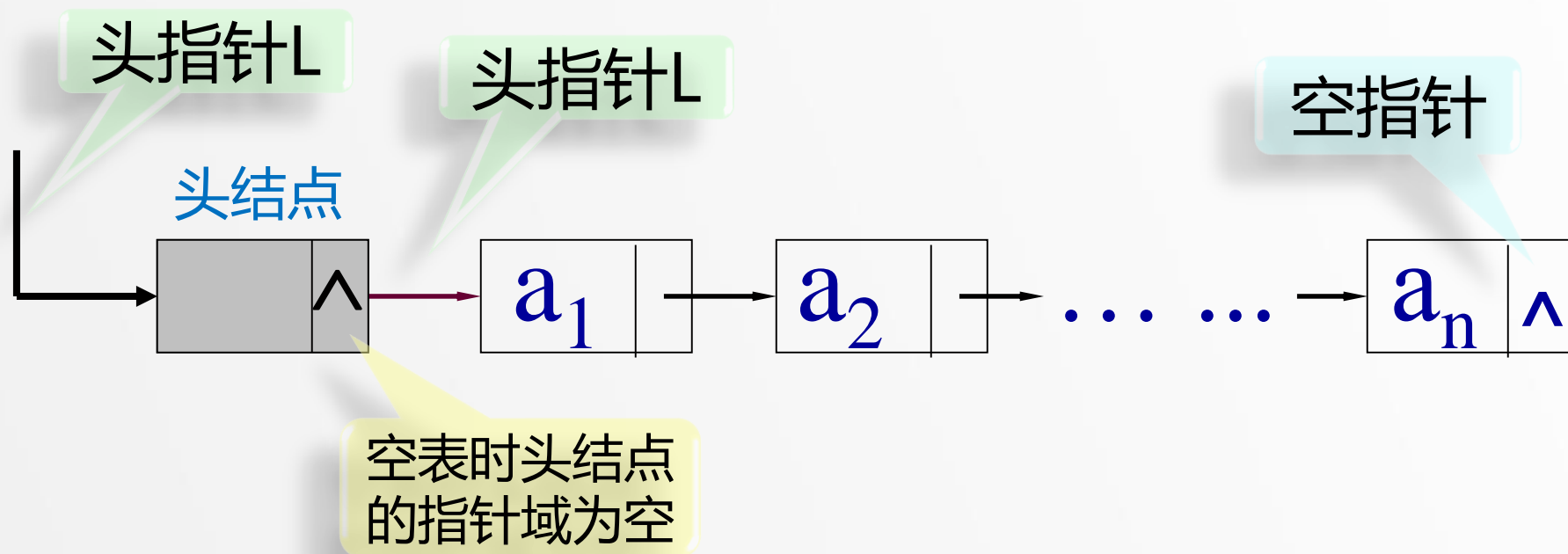


- 物理结构

■ 例如：  
( 'a' , ' b' , ' c' , ' d' , ' e' , ' f' , ' g' )  
物理结构(一种)





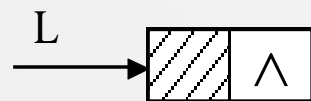


以线性表中第一个数据元素  $a_1$  的存储地址作为线性表的地址，称作线性表的头指针。

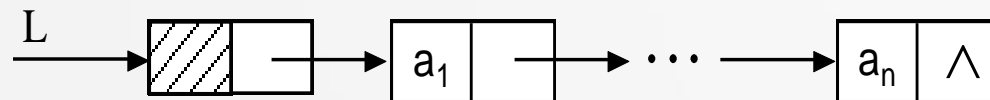
有时为了操作方便，在第一个结点之前虚加一个“头结点”，并用链表的头指针指向头结点，称为带头结点的单链表。



- 不带头结点
  - 链表指针存放链表第一个数据元素结点的地址
  - 空链表时该指针域为NULL
- 带头结点的单链表
  - 一个专门的结点，称为头结点
  - 该头结点永远存在
  - 该头结点指针域存放第一个数据元素结点的地址
  - $L.next = NULL$



(a)空表



(b)非空表

## 练习题

1. 线性表可以采用顺序存储结构存储数据，也可以采用链式存储结构存储数据？
2. 访问线性表的第 $i$ 个元素，顺序存储结构通过地址计算公式可以一步到位访问到，采用单链表需要从头指针开始依次访问单链表的每一个元素，最坏时间复杂度为 $O(n)$
3. 链式存储结构不止单链表一种链表形态来存储线性表的元素和关系。

## 练习题

1. 头指针为L，则单链表为空的表示方法为 $L = \text{NULL}$ ?
2. 已知指针p指向单链表L的第i个结点，则访问第i个数据和第i+1个数据的方法分别是 $p \rightarrow \text{data}$ 和 $p \rightarrow \text{next} \rightarrow \text{data}$