

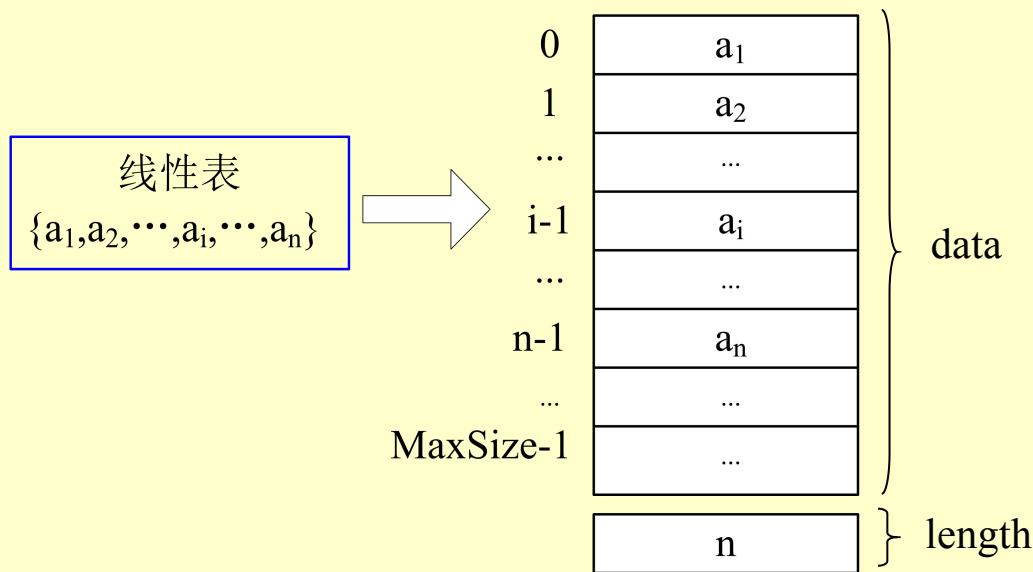


本节主题:

线性表的顺序存储结构

# 线性表的顺序存储—顺序表

- 把线性表中的所有元素按照其逻辑顺序依次存储到从计算机存储器中指定存储位置开始的一块连续的存储空间中。



占用空间： $n * \text{sizeof}(\text{ElemType})$

- 元素类型为ElemType

- 线性表的长度为n

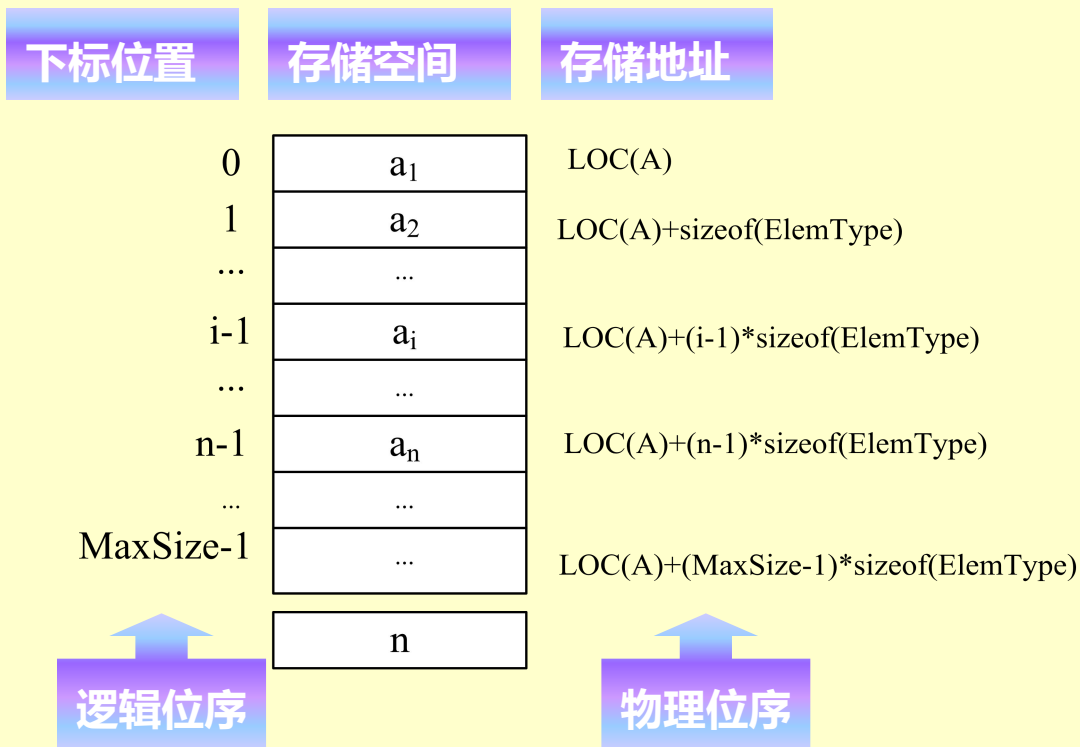
区别

- 线性表（逻辑结构）

- 顺序表（存储结构）

# 顺序表存储类型

```
#define MaxSize 50
typedef struct
{
    ElemType data[MaxSize];
    int length;
} SqList;
```



## 例：城市表的顺序存储

```
#define MaxSize 50
typedef struct city
{
    char code[4];
    char name[10];
    char describe[16];
} ElemType;
typedef struct
{
    ElemType data[MaxSize];
    int length;
} SqList;
```

存储地址

100

130

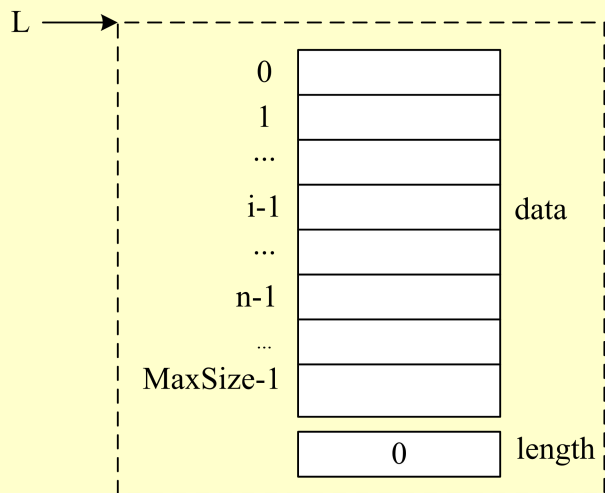
160

190

210

区号	城市名	说明
010	Beijing	首都
021	Shanghai	直辖市
027	Wuhan	湖北省省会
029	Xian	陕西省省会
025	Nanjing	江苏省省会

## 接下来的话题：如何实现基本运算？



- ( 1 ) 初始化线性表  $\text{InitList}(\&L)$
- ( 2 ) 销毁线性表  $\text{DestroyList}(\&L)$
- ( 3 ) 判线性表是否为空表  $\text{ListEmpty}(L)$
- ( 4 ) 求线性表的长度  $\text{ListLength}(L)$
- ( 5 ) 输出线性表  $\text{DispList}(L)$
- ( 6 ) 求线性表  $L$  中指定位置的某个数据元素  $\text{GetElem}(L, i, \&e)$
- ( 7 ) 查找元素  $\text{LocateElem}(L, e)$
- ( 8 ) 插入元素  $\text{ListInsert}(\&L, i, \&e)$
- ( 9 ) 删除元素  $\text{ListDelete}(\&L, i, \&e)$