

线性表的基本操作

```
Status List_Init(SqListPtr L);  
void List_Destroy(SqListPtr L);  
void List_Clear(SqListPtr L);  
bool List_Empty(SqListPtr L);  
int List_Size(SqListPtr L);  
Status List_Retrial(SqListPtr L, int pos, ElemType *elem);  
Status List_Locate(SqListPtr L, ElemType elem, int *pos);  
Status List_Insert(SqListPtr L, int pos, ElemType elem);  
Status List_delete(SqListPtr L, int pos);  
Status List_Prior(SqListPtr L, int pos, ElemType * elem);  
Status List_Next(SqListPtr L, int pos, ElemType *elem);
```

合并线性表算法

```
Status List_Union (SqListPtr La, SqListPtr Lb){
    ElemType elem;                /* 存放从Lb中取出的元素*/
    Status status;                /* 状态代码*/
    int i, j, len = List_Size(Lb); /* len存放Lb的元素个数*/
    for (i=1; i<=len; i++){
        List_Retrieve(Lb, i, &elem); /* 取出Lb中第i个数据元素*/
        status = List_Locate(La, elem, &j); /* 判它是否在La中*/
        if(status!= success){           /* 如果不在*/
            status = List_Insert(La, 1, elem); /* 插入到第一个位置*/
            if(status!= success) break;      /* 插入失败则退出*/
        }
    }
    return status;
}
```

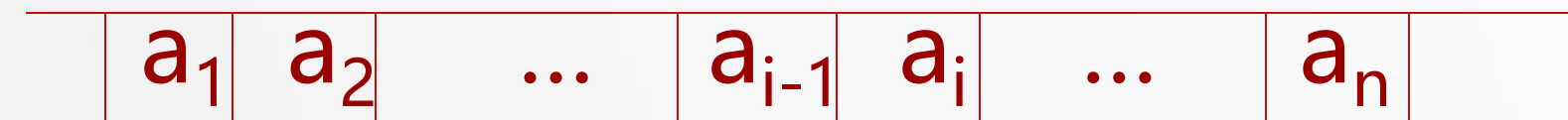
CONTENTS

目录

- 1 线性表的定义
- 2 线性表的抽象数据类型
- 3 线性表的顺序存储结构**
- 4 线性表的链式存储结构
- 5 顺序与链式存储结构比较
- 6 线性表的简单应用
- 7 线性表的扩展

线性表的顺序存储结构

用一组地址连续的存储单元
依次存放线性表中的数据元素



线性表的起始地址s
称作线性表的基地址

可以随机存取数据

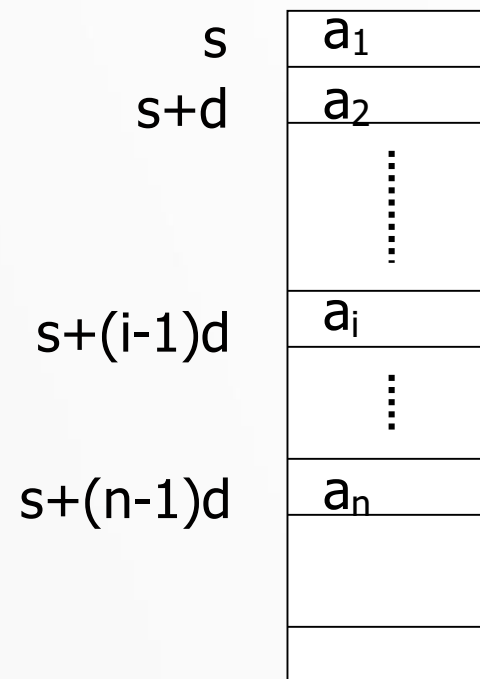
以 “存储位置相邻” 表示有序对 $\langle a_{i-1}, a_i \rangle$

$$\text{即: } \text{LOC}(a_i) = \text{LOC}(a_{i-1}) + d$$

一个数据元素所占存储单元↑

数据元素 a_i 的存储位置为:

$$\text{LOC}(a_i) = \underbrace{\text{LOC}(a_1)}_{\substack{\uparrow \text{基地址} \\ s}} + (i-1) \times d$$



讨论:

题目：如果线性表有 n 个元素 $L=(a_1, a_2, \dots, a_n)$ ，如何用某种程序设计语言，比如C语言存储这 n 个数据并实现建立线性表、查找线性表的某个数据，在某个位置插入或者删除一个数据等基本操作？这些基本操作实现之后，更复杂的操作，比如合并线性表是不是就能够通过直接调用这些基本操作实现？并说明理由。