
数据结构

顺序表的实现

创客学院 小美老师

线性表的基本运算

线性表的基本运算

设线性表 $L=(a_0, a_1, \dots, a_{n-1})$, 对 L 的基本运算有:

1) 建立一个空表: `list_create(L)`

2) 置空表: `list_clear(L)`

线性表

3)判断表是否为空: $\text{list_empty}(L)$ 。若表为空, 返回值为1, 否则

返回 0

4)求表长: $\text{length}(L)$

5)取表中某个元素: $\text{GetList}(L, i)$, 即 a_i 。要求 $0 \leq i \leq \text{length}(L)-1$

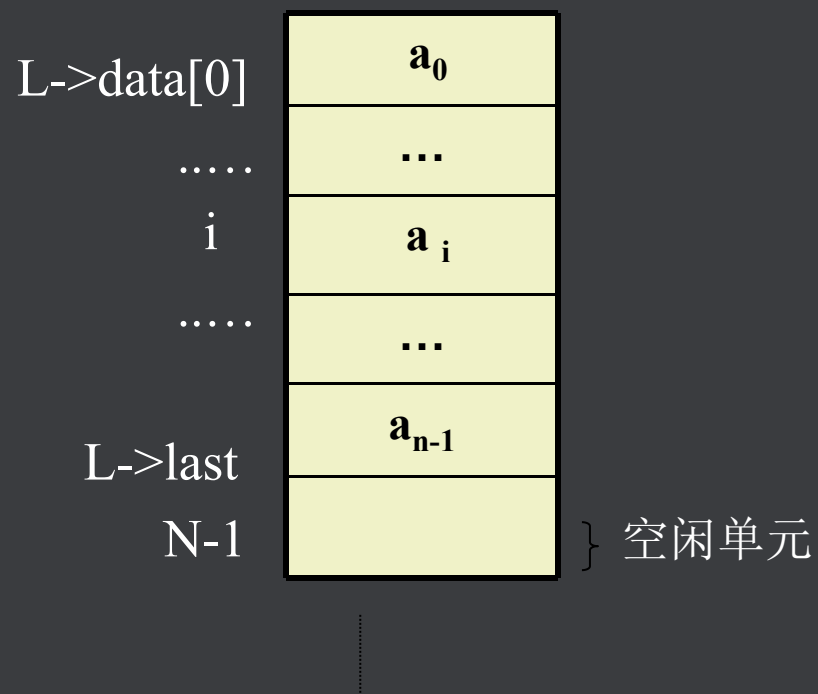
6)定位运算: $\text{Locate}(L, x)$ 。确定元素 x 在表 L 中的位置 (或序号)

$$\text{Locate}(L, x) = \begin{cases} i & \text{当元素 } x = a_i \in L, \text{ 且 } a_i \text{ 是第一个与 } x \text{ 相等时;} \\ -1 & x \text{ 不属于 } L \text{ 时。} \end{cases}$$

基本运算的相关算法

定位：确定给定元素 x 在表 L 中第一次出现的位置（或序号）。

即实现 $\text{Locate}(L, x)$ 。算法对应的存储结构如图所示。



线性表

7)插入:

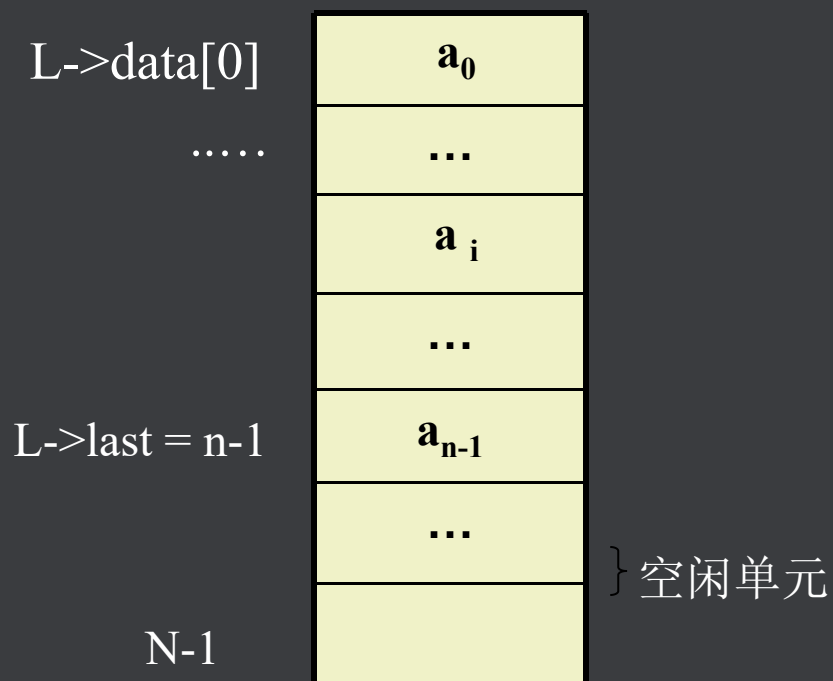
Insert(L,x,i)。将元素x插入到表L中第i个元素 a_i 之前,且表长+1。

插入前: $(a_0, a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_{n-1})$ $0 \leq i \leq n$, $i=n$ 时,x插入表尾

插入后: $(a_0, a_1, \dots, a_{i-1}, x, a_i, a_{i+1}, \dots, a_{n-1})$

基本运算的相关算法

- 算法思路：若表存在空闲空间，且参数*i*满足： $0 \leq i \leq L \rightarrow \text{last} + 1$ ，则可进行正常插入。
插入前，将表中（ $L \rightarrow \text{data}[L \rightarrow \text{last}] \sim L \rightarrow \text{data}[i]$ ）部分顺序下移一个位置，然后将 *x* 插入 $L \rightarrow \text{data}[i]$ 处即可。算法对应的表结构。



线性表

8)删除:

Delete(L,i)。删除表L中第i个元素 a_i , 且表长减1, 要求 $0 \leq i \leq n-1$ 。

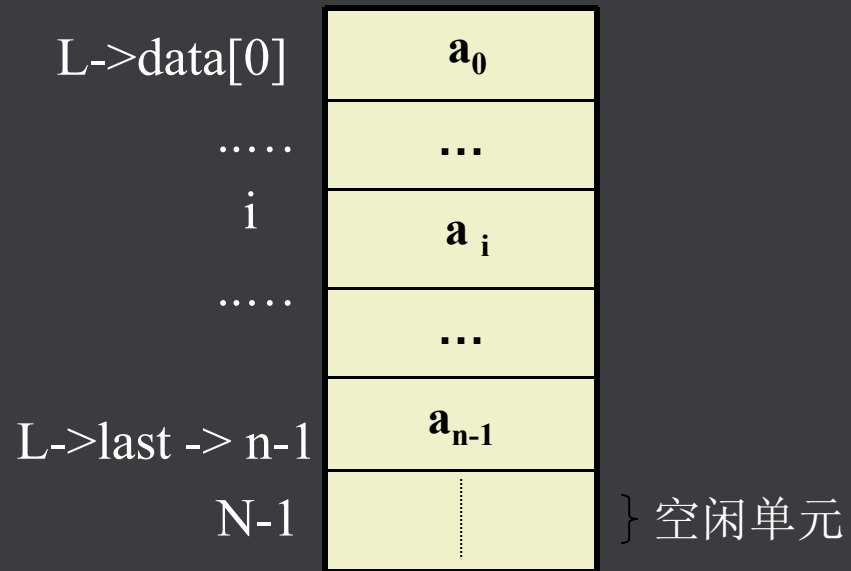
删除前: $(a_0, a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_{n-1})$

删除后: $(a_0, a_1, \dots, a_{i-1}, a_{i+1}, \dots, a_n)$

基本运算的相关算法

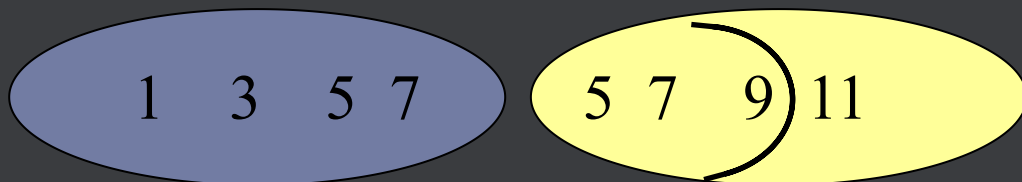
删除：将表中第 i 个元素 a_i 从表中删除，即实现DeleteSqlist(L, i)。

算法思路：若参数 i 满足： $0 \leq i \leq L \rightarrow \text{last}$ ，将表中 $L \rightarrow \text{data}[i+1] \sim L \rightarrow \text{data}[L \rightarrow \text{last}]$ 部分顺序向上移动一个位置，覆盖 $L \rightarrow \text{data}[i]$ 。



线性表的基本运算

设线性表 $La=(a_0a_1, \dots, a_{m-1})$, $Lb=(b_0b_1, \dots, b_{n-1})$, 求 $La \cup Lb$
 $\Rightarrow La$,



算法思路：依次取表 Lb 中的 $b_i(i=0,1,\dots,n-1)$, 若 b_i 不属于 La ,则将其插入表 La 中。

线性表

设计清除线性表 $L=(a_0, a_1, \dots, a_i, \dots, a_{n-1})$ 中重复元素的算法。

算法思路：对当前表 L 中的每个 a_i ($0 \leq i \leq n-2$)，依次与 a_j ($i+1 \leq j \leq n-1$)

比较，若与 a_i 相等，则删除之。

线性表的顺序存储缺点

线性表的顺序存储结构有存储密度高及能够随机存取等优点，但存在以下不足：

(1)要求系统提供一片较大的连续存储空间。

(2)插入、删除等运算耗时，且存在元素在存储器中成片移动的现象；

扫一扫，获取更多信息



THANK YOU