**1. 线性结构定义**

线性结构：表示数据元素之间的(广义的)**有序关系**

包括：**线性表、栈、队列、串、广义表**

特点： 存在唯一一个“第一个“和”最后一个“元素

除第一个外，每个元素均有唯一一个直接前驱

除最后一个外，每个元素均有唯一一个直接后继

线性表的长度：数据元素的个数n

空表：n=0

非空表：第一个节点：首结点

最后一个结点：尾结点

有序表：若线性表中结点按值或按关键字值大小顺序排列的，称线性表是**有序的**

**基本操作**

|  |  |
| --- | --- |
| 操作 | 最坏复杂度 |
| GetElem(L, i, &e) //取线性表第i个元素 | n |
| LocateElem(L, e, compare) //定位数据元素 | n |
| ListInsert ( &L, i, &e ) //插入数据元素 | n |
| ListTraverse(L, visit) //遍历线性表 | n |

|  |  |  |
| --- | --- | --- |
| 复合操作 | 涉及操作 | 最坏复杂度 |
| union (List &La, List Lb) //合并 | ListLength、GetElem、LocateElem、ListInsert | mn2 或m2n  n3 |
| multiset(List &La, List Lb) // 去重 | InitList(La)、长度、获取、定位、插入 |
| MergeList (La, Lb, &Lc) //合并(非顺序表) |  | m2+n2 |

**2. 线性表的顺序表示和实现**

**数据结构SqList：**

elem：数据**数组基地址**、或动态分配空间的指针

listsize：数组长度/当前分配的存储容量

length：当前已有结点数

顺序表操作：

删除=删除指定数据+移动后面数据 O(n)

合并顺序表：O(La->length+Lb->length)

**3. 线性表的链式表示和实现**

**单链表LinkedList：**

**head，tail：指向头节点和最后一个节点**

**current：指向当前被访问的结点，初始指向头节点**

**int curpos：//指示当前指针位置,初值为0**

**len：链表长度**

**Lnode：**data数据

next指向下一个结点

操作注意： 将值为e的新结点插入到表的第i个结点的位置上

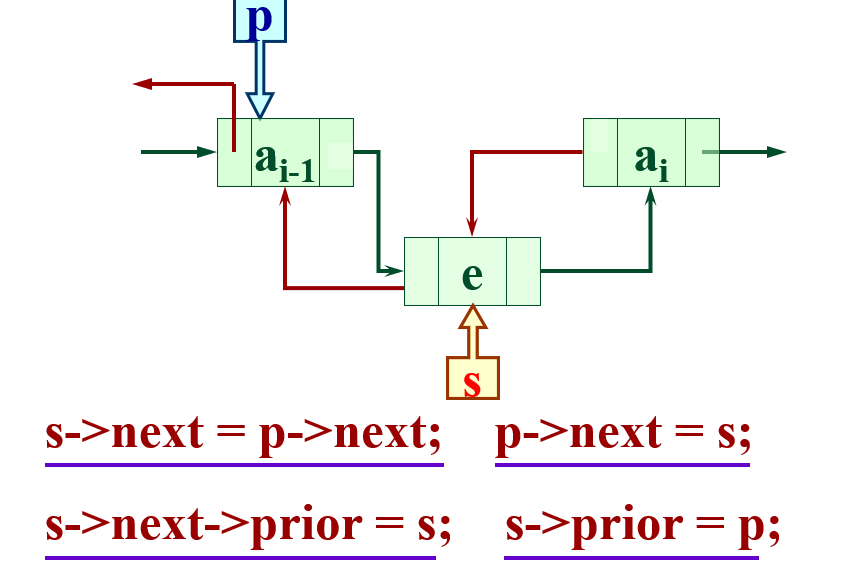
首先找到**第i-1个结点**p

**双向链表DoublyLinkedList：与node结构相同，但是它是一个头节点**

**node：** data数据

node \*prior，\*next：分别指向该节点的前驱和后继

**插入操作时注意操作顺序**



**循环链表CircularLinkedList**

单链循环链表：最后一个结点的指针域指向头结点

双链循环链表：**头节点前驱为空**，最后一个结点后继头结点