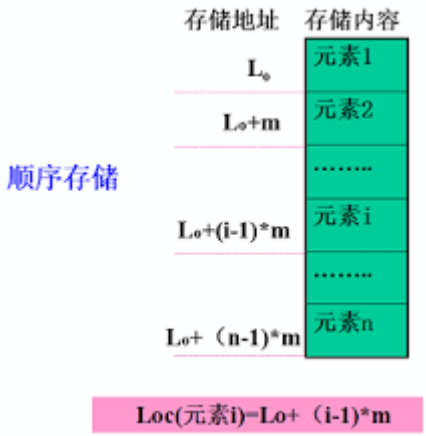
线性表的顺序表示和实现

线性表的顺序表示指的是用一组地址连续的存储单元依次存储线性表的数据元素



一般来说，线性表的第i个元素ai的存储位置为

LOC(ai) = LOC(a1) + (i - 1) * l

LOC(a1)是线性表的第一个数据元素a1的存储位置，通常称作线性表的起始位置或基地址

由上公式可知，每个数据元素的存储地址都和线性表的起始地址相差其在线性表位序成正比的常数

结构体定义

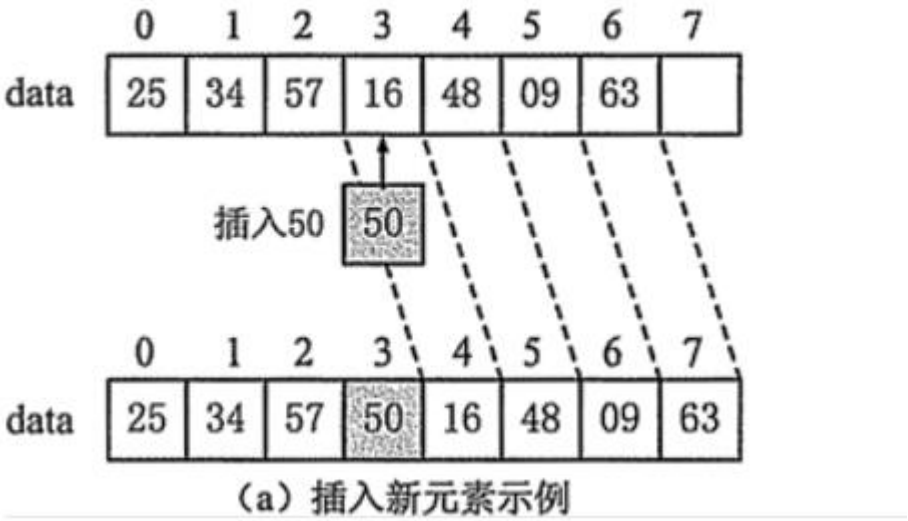
```
typedef struct {
    ElementType  *elements; //存储地址基地址
    int          length;    //当前长度
    int          listSize;  //存储容量
} ArrayList;
```

顺序表的插入

线性表的插入操作就是指在线性表的第i-1个数据元素和第i个数据元素之间插入一个新的数据元素

- (a1, ..., ai-1, ai, ..., an)
- 在位序i，插入元素b
- (a1, ..., ai-1, b, ai, ..., an)

就是要让表长为n的线性表变为表长为n+1的线性表



（a）插入新元素示例

因为在顺序表中，由于逻辑上相邻的数据元素在物理位置上也是相邻的，所以在插入数据元素时必须移动数据元素才能反映这个变化；

一般情况下，在第i个元素之间插入一个元素时，需将第n至第i（共n-i+1）个元素向后移动一个位置

Code

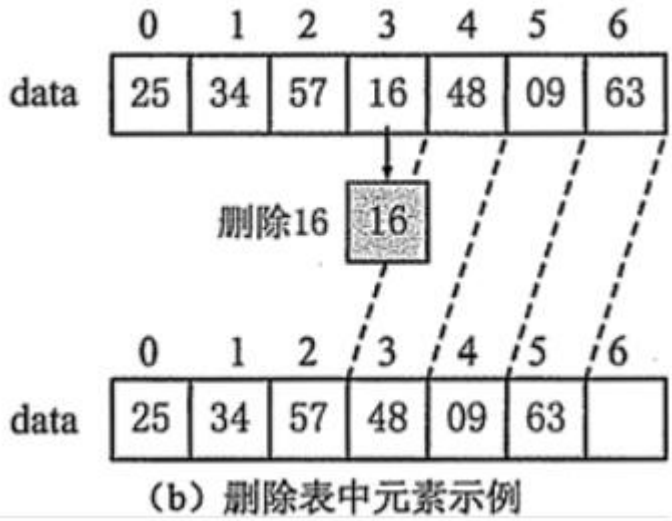
```
//要插入的位置之后 每个元素后移一位
//必须采用从后开始的方式 从前开始的话元素会被覆盖
for (int i = list.length - 1;i >= index;i--){
    list.element[i + 1] = list.element[i];
}

//插入新元素
list.element[index] = newElement;

//表长加1
list.length++;
```

插入时间复杂度：O(n)

插入平均移动次数：n/2

顺序表的删除

一般情况下，删除第i个元素时需要将从第i+1个元素至第n（共n-i）个元素依次向前移动一个位置



（b）删除表中元素示例

Code

```
//索引index之后的元素全都前移一位
//必须采用从前开始的方式 从后开始的话元素会被覆盖
for (int i = index;i < list.length - 1;i++){
    list.element[i] = list.element[i + 1];
}

//表长减1
list.length--;
```

删除时间复杂度：O(n)

删除平均移动次数：(n-1)/2