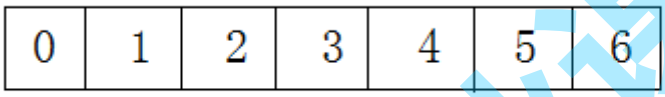
2--线性表之-顺序表

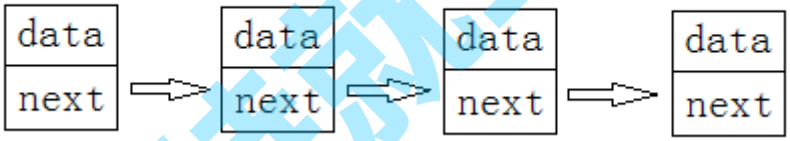
【本节目标】

- **1.线性表概念**
- **2.顺序表实现**
- **3.顺序表相关OJ题练习**

1.线性表

> 线性表 *(linear list)* 是n个具有相同特性的数据元素的有限序列。 线性表是一种在实际中广泛使用的数据结构，常见的线性表：顺序表、链表、栈、队列、字符串...
>
> 线性表在逻辑上是线性结构，也就说是连续的一条直线。但是在物理结构上并不一定是连续的，线性表在物理上存储时，通常以数组和链式结构的形式存储。





2.顺序表实现

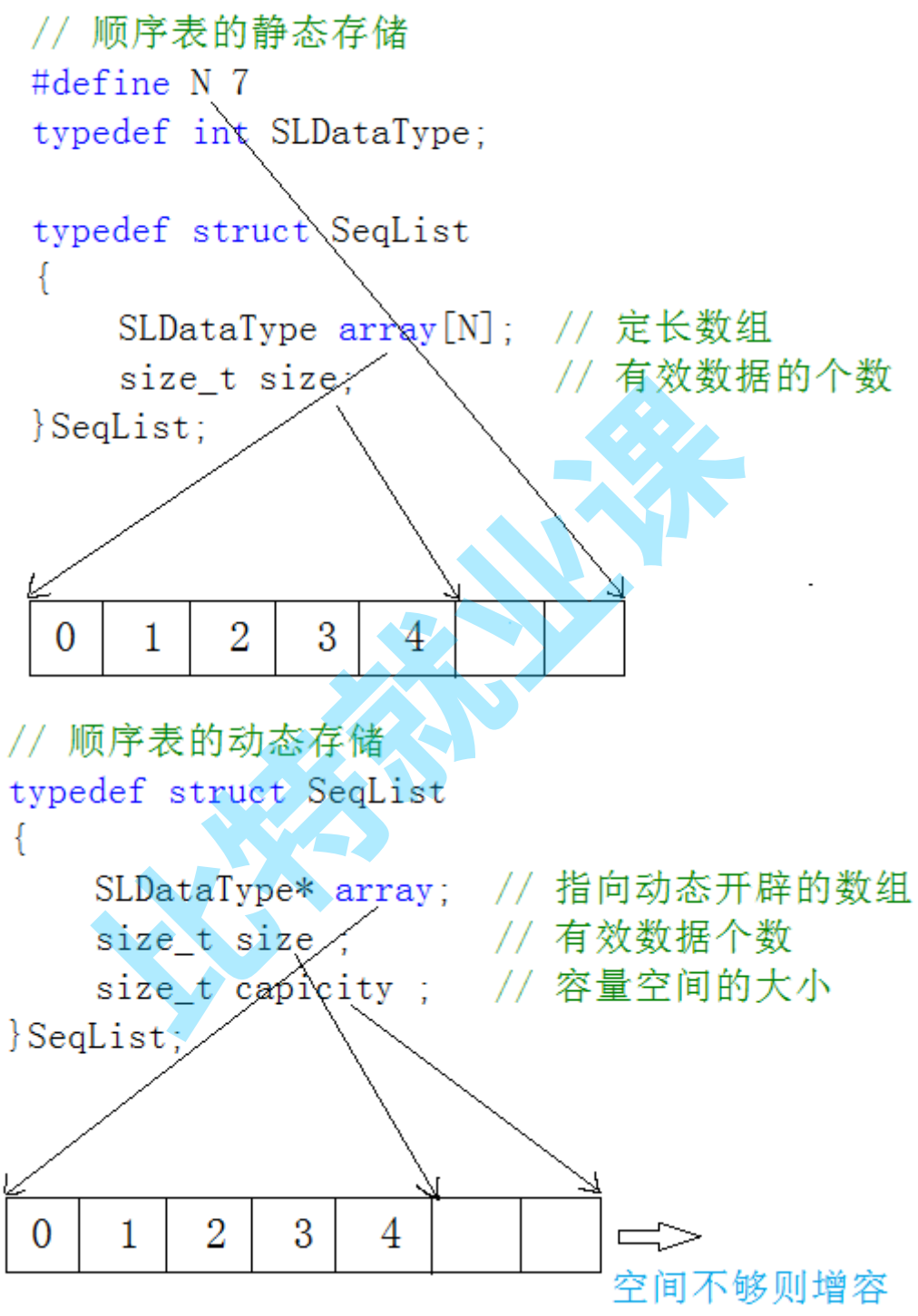
> ### 2.1概念及结构
>
> 顺序表是用一段**物理地址连续**的存储单元**依次存储数据元素**的线性结构，一般情况下采用数组存储。在数组上完成数据的增删查改。
>
> 顺序表一般可以分为：
>
> 1. 静态顺序表：使用定长数组存储。
> 2. 动态顺序表：使用动态开辟的数组存储。

```
// 顺序表的静态存储
#define N 100
typedef int SLDataType;

typedef struct SeqList
{
     SLDataType array[N]; // 定长数组
     size_t size;         // 有效数据的个数
}SeqList;


// 顺序表的动态存储
typedef struct SeqList
```

```
{
    SLDataType* array;  // 指向动态开辟的数组
    size_t size ;       // 有效数据个数
    size_t capicity ;   // 容量空间的大小
}SeqList;
```



2.2 接口实现：

静态顺序表只适用于确定知道需要存多少数据的场景。静态顺序表的定长数组导致N定大了，空间开多了浪费，开少了不够用。所以现实中基本都是使用动态顺序表，根据需要动态的分配空间大小，所以下面我们实现动态顺序表。

```
// 顺序表的动态存储
typedef struct SeqList
{
```

```
    SLDataType* array;  // 指向动态开辟的数组
    size_t size ;       // 有效数据个数
    size_t capicity ;   // 容量空间的大小
}SeqList;

// 基本增删查改接口
// 顺序表初始化
void SeqListInit(SeqList* psl);
// 顺序表销毁
void SeqListDestory(SeqList* psl);
// 顺序表打印
void SeqListPrint(SeqList* psl);
// 检查空间，如果满了，进行增容
void CheckCapacity(SeqList* psl);
// 顺序表尾插
void SeqListPushBack(SeqList* psl, SLDataType x);
// 顺序表尾删
void SeqListPopBack(SeqList* psl);
// 顺序表头插
void SeqListPushFront(SeqList* psl, SLDataType x);
// 顺序表头删
void SeqListPopFront(SeqList* psl);
// 顺序表查找
int SeqListFind(SeqList* psl, SLDataType x);
// 顺序表在pos位置插入x
void SeqListInsert(SeqList* psl, size_t pos, SLDataType x);
// 顺序表删除pos位置的值
void SeqListErase(SeqList* psl, size_t pos);
```

3. 顺序表相关OJ题练习

3.1 原地移除元素：https://leetcode-cn.com/problems/remove-element/

示例 2：

```
输入：nums = [0,1,2,2,3,0,4,2], val = 2
输出：5, nums = [0,1,4,0,3]
解释：函数应该返回新的长度 5, 并且 nums 中的前五个元素为 0, 1, 3, 0, 4。注意这五个元素可为任意顺序。你不需要考虑数组中超出新长度后面的元素。
```

思路二：开辟一个同样大小的数组tmp，把不是2的数据放到tmp，再把tmp中的数据拷贝回来。时间复杂度O(N)，空间复杂度位O(N)

```
nums = [0,1,2,2,3,0,4,2], val = 2

tmp = [0 1 3 0 4      ]
```

思路三：在思路二的基础上，不开辟额外的数组tmp，直接把不是2的数据从开始放到原数组的位置

```
          src
nums = [0,1,2,2,3,0,4,2], val = 2
          dst

               src
nums = [0,1,2,2,3,0,4,2], val = 2
               dst
```

3.2 合并两个有序数组：https://leetcode-cn.com/problems/merge-sorted-array/