

数据类型和抽象数据类型

抽象数据类型 — ADT — 对象，关系，运算

- 线性结构：元素之间满足**线性关系**

线性表：

- 相同特性 — 相同的数据类型
- 有穷
- 序列

基本运算：创建空表，插入元素，删除元素，查找元素，判别

存储结构：顺序存储，链式存储

STL

- STL容器：顺序容器，关联容器，容器适配器

顺序容器

顺序容器：元素在容器中的值与位置无关，即容器是**不排序的**

- vector, deque：变长的数组结构
- list：双向链表

顺序容器的成员函数 - PPT2-STL初步, page 5

函数：size, empty, front, back, push_back, pop_back, clear

vector

- vector表示一个**可变长数组**。在预分配的存储空间存满以后，再插入新数据时，**存储空间自动翻倍**> vector内存增长机制 PPT2-STL初步, page 12
- 所有数据元素保存在一块**连续内存中**
- **中间位置**插入或删除元素的速度**很慢**
- push_back()和pop_back()都很**高效**，适合在表头和表尾操作

```
vector<int> va;  
vector<va> va(20);  
  
va.capacity()// 内存大小
```

线性表的顺序表示和实现（顺序表）

定义

- 计算机内存中的“物理位置相邻”表示逻辑关系
- 顺序表为“随机存取（random access，直接存取）的存储结构”：也就是说，访问到任意位置的元素所花的时间相同
- 随机存取——顺序存取
- typedef：声明新的类型名来代替已有的类型名（大写字母）
- 版本一：

```
#define MAXNUM 100
struct SeqList
{
    DataType element[MAXNUM];
    int n; // 顺序表中当前元素的个数
};
typedef struct SeqList *PSeqList;
PSeqList pa;
```

使用指针指向的位置来访问

- 版本二：element作为地址
- 版本三：加入了MAXNUM的定义

插入

PPT2-线性表1, page 52

- 先判断存储够不够（是否溢出）
- 插入点处的元素依此右移
- 插入

时间复杂性：与插入位置和表长有关，长度为n的线性结构中在下标为i的元素插入一个元素需移动元素的次数为(n-i)；一共有n+1个位置来插入

$$E_{is} = \sum_{i=0}^n p_i (n - i) = \frac{n}{2}$$

删除

- 判断是否存在这个下标
- 删除
- 前移

可以删除的是n个，时间复杂性是 $\frac{n-1}{2}$

应用