

【数据结构】 Day3

▼ Class	Advanced Data Structures
📅 Date	@December 5, 2021
🔗 Material	
## Series Number	
☰ Summary	

【Ch3】表、栈和队列

3.2 表ADT

我们将处理一般的形如 $A_1, A_2, A_3 \dots, A_N$ 的表，我们说这个表的**大小是N**。我们称**大小为0的表为空表**

我们需要定义一些在表ADT上进行的操作的集合：

1. PrintList和MakeEmpty是常用的操作
2. Find返回关键字首次出现的位置
3. Insert和Delete一般是从表的某个位置插入和删除某个关键字
4. FindKth则返回某个位置上的元素

3.2.1 表的简单数组实现

对表的所有操作都**可以通过数组来实现**。通常数组需要估计得大一些，这样会浪费大量的空间。这是最严重的局限，特别是存在许多未知大小的表的情况下

数组实现使得PrintList和Find正如所预期的那样以线性时间执行，而FindKth则花费常数时间。

然而，**插入和删除的花费是昂贵的(因为插入和删除一个元素需要移动目标元素之前或之后的所有元素)**。最坏的情况均为 $O(N)$

3.2.2 链表

为了避免插入和删除的线性开销，我们需要允许表可以不连续存储。否则表的部分或全部需要整体移动

链表由一系列不必在内存中相连的结构组成。每一个结构均含有表元素和指向包含该元素后继元的结构的指针。我们称之为Next指针。最后一个单元的Next指针指向NULL，该值由C定义并且不能够与其他指针混淆。ANSI C规定NULL为0

3.2.4 常见的错误

最常遇到的错误是你的程序因来自系统的棘手的错误信息而崩溃，比如"memory access violation"或"segmentation violation"。这种信息通常意味着有指针包含了伪地址