栈与队列

# 栈

栈（Stack）是一种常见的数据结构，它遵循后进先出（Last In First Out，LIFO）的原则。栈的基本操作包括入栈（压栈，Push）和出栈（弹栈，Pop），以及其他一些辅助操作如查看栈顶元素等。

栈的具体实现可以使用数组或链表等数据结构。在数组实现中，栈底通常位于数组的起始位置，栈顶则随着元素的入栈和出栈操作而动态变化。在链表实现中，栈顶元素通常位于链表的头部，便于快速进行入栈和出栈操作。

栈在计算机科学中有广泛的应用，如函数调用栈、表达式求值、括号匹配等问题都可以利用栈来解决。此外，栈还在算法设计、操作系统、编译器设计等领域发挥着重要作用。

# 队列

队列（Queue）也是一种常见的数据结构，它遵循先进先出（First In First Out，FIFO）的原则。队列的基本操作包括入队（在队尾添加元素）和出队（在队头删除元素），以及其他一些辅助操作如查看队头元素等。

队列的具体实现可以使用数组或链表等数据结构。在数组实现中，通常需要维护两个指针，分别指向队头和队尾，以便高效地进行入队和出队操作。在链表实现中，队头元素通常位于链表的头部，队尾元素位于链表的尾部，这样可以方便地在队尾添加元素和从队头删除元素。

队列在计算机科学中有广泛的应用，如多线程编程中的任务队列、缓冲区的处理、广度优先搜索等算法的实现等。队列还可以用于模拟现实生活中的一些场景，如排队购物、排队打印等。

与栈相比，队列的主要区别在于它们的操作顺序不同。栈是后进先出（LIFO），而队列是先进先出（FIFO）。这意味着在栈中，最后进入的元素会首先被取出；而在队列中，最先进入的元素会首先被取出。