数据结构与算法

# 数组与简单排序

## 冒泡排序

冒泡排序（Bubble Sort）是一种简单的排序算法。它重复地走访过要排序的数列，一次比较两个元素，如果他们的顺序错误就把他们交换过来。走访数列的工作是重复地进行直到没有再需要交换，也就是说该数列已经排序完成。这个算法的名字由来是因为越小的元素会经由交换慢慢“浮”到数列的顶端。

冒泡排序算法的运作如下:

1. 比较相邻的元素。如果第一个比第二个大，就交换他们两个。
2. 对每一对相邻元素作同样的工作，从开始第一对到结尾的最后一对。在这一点，最后的元素应该会是最大的数。
3. 针对所有的元素重复以上的步骤，除了最后一个。
4. 持续每次对越来越少的元素重复上面的步骤，直到没有任何一对数字需要比较。

## 选择排序

选择排序是常用内部排序的一种，常见的实现算法有直接选择排序算法和堆排序算法，选择排序的基本思想是每次从待排数据中选择第n小的数据放到排序列表的第n个位置，假如共有N个数据待排，那么经过N-1次排序后，待排数据就已经按照从小到大的顺序排列了。

　　直接选择排序算法的思想比较简单：（假设数据放在一个数组a中，且数组的长度是N）

　　1：从a[0]-a[N-1]中选出最小的数据，然后与a[0]交换位置

　　2：从a[1]-a[N-1]中选出最小的数据，然后与a[1]交换位置（第1步结束后a[0]就是N个数的最小值）

　　3：从a[2]-a[N-1]中选出最小的数据，然后与a[2]交换位置（第2步结束后a[1]就是N-1个数的最小值）

　　以此类推，N-1次排序后，待排数据就已经按照从小到大的顺序排列了

## 插入排序

不扯太多概念性的东西，简单点来说，插入排序 将数组所有元素划分成了有序区和无序区，假设当前数组有 N 个元素，  
开始默认第一个元素（下标为0）所处的位置是有序区，这是局部有序，从第二个元素（i=1）至数组最后一个元素（i=N-1）属于无序区；  
假设数组元素是按从左至右存放的，如果用 i 来标记无序区中的第一个元素下标，也就是无序区中最左边或者说是无序区中下标值最小的下标，  
则每趟排序是将下标 i 所指向的有效值插入有序区的适当位置，使得每趟排序完成之后，有序区的所有元素总是保持局部有序状态。  
按这样来回 N -1 趟插入之后，则 N 个元素已成有序状态。  
  
尽管插入排序的复杂度也是 O(n^2)，但一般情况下，插入排序会比冒泡排序快一倍，要比选择排序还要快一点。

# 栈与队列

通常称,栈和队列是限定插入和删除只能在表的’端点’进行的线性表

栈和队列是俩种操作受限的线性表,是俩种常用的数据类型.

## 栈

### 定义

* 栈是仅限制在表尾进行插入和删除操作的特殊线性表,限制操作的表尾端称为”栈顶”,另一端称为”栈底”
* 栈是后进先出的线性表(LIFO)，或者是先进后出(FILO)的线性表

### 基本操作

* 栈的置空操作: clear()
* 栈的判空操作: isEmpty()
* 求栈的长度: length()
* 取栈顶元素操作: peek()
* 入栈操作: push()
* 出栈操作:pop()

### 顺序栈

### 链栈

栈的链式存储结构称为链栈,是运算受限的单链表,其插入和删除操作仅限在链表的表头位置上进行,故链栈没有必要向单链表一样附加头结点,栈顶指针即为链表的头指针.

### 栈的应用

1. 分隔符匹配问题

算法设计思想:

1. 凡出现左括弧,则进栈
2. 凡出现右括弧,首先检查栈是否为空,若栈空则表明该右括弧多余

否则和栈顶元素比较，若相匹配,则左括弧出栈,否则表面不相匹配

c.表达式检验结束时,若栈空则表明表达式中匹配正确,否则表面左括弧有余