# **数据结构与算法**

最常用的数据结构与算法：

数据结构：数组、链表、栈、队列、散列表、二叉树、堆、跳表、图、Tire树

算法：递归、排序、二分查找、搜索、哈希算法、贪心算法、分治算法、回溯算法、动态规划、字符串匹配算法。

1. 排序
2. 排序算法与复杂度归类
3. 几种最经典最常用的排序方法：冒泡排序、插入排序、选择排序、快速排序、归并排序、计数排序、基数排序、桶排序。
4. 复杂度归类

冒泡排序、插入排序、选择排序O(n^2)

快速排序、归并排序、堆排序O(nlogn)

计数排序、基数排序、桶排序O（N）

二、如何分析一个“排序算法”？

<1>算法的执行效率

1. 最好、最坏、平均情况时间复杂度。

2. 时间复杂度的系数、常数和低阶。

3. 比较次数，交换（或移动）次数。

<2>排序算法的稳定性

1. 稳定性概念：如果待排序的序列中存在值相等的元素，经过排序之后，相等元素之间原有的先后顺序不变。

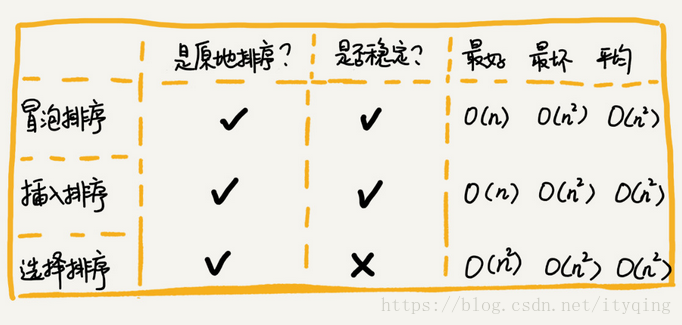
2. 稳定性重要性：可针对对象的多种属性进行有优先级的排序。

3. 举例：给电商交易系统中的“订单”排序，按照金额大小对订单数据排序，对于相同金额的订单以下单时间早晚排序。用稳定排序算法可简洁地解决。先按照下单时间给订单排序，排序完成后用稳定排序算法按照订单金额重新排序。

<3>排序算法的内存损耗

原地排序算法：特指空间复杂度是O(1)的排序算法。

常见的排序算法：

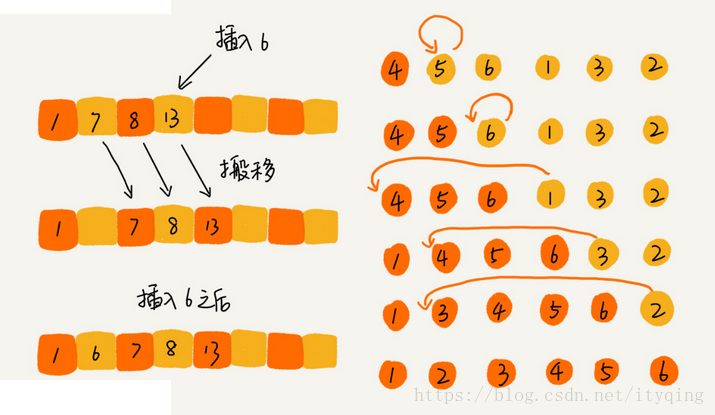


**冒泡排序**



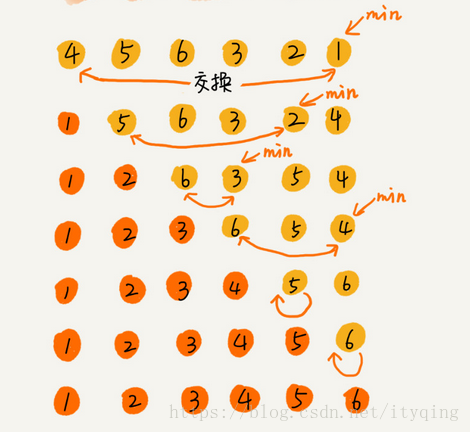
冒泡排序只会操作相邻的两个数据。每次冒泡操作都会对相邻的两个元素进行比较，看是否满足大小关系要求，如果不满足就让它俩交换。

**插入排序**



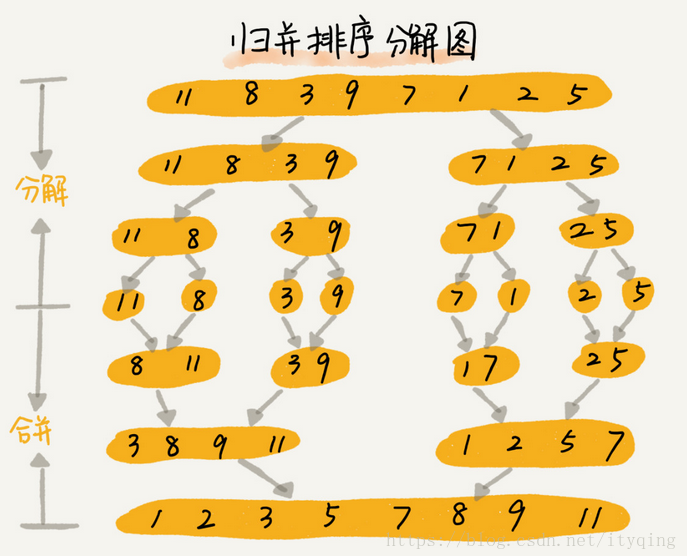
插入排序将数组数据分成已排区间和未排序区间。初始已排序区间只有一个元素，即数组第一个元素。在未排序区间取出一个元素插入到已排序区间的合适位置，直到未排序区间为空。

**选择排序**



选择排序将数组分成已排序区间和未排序区间。初始已排序区间为空。每次从未排序区间中选出最小的元素插入已排序区间的末尾，直到未排序区间为空。

**归并排序**

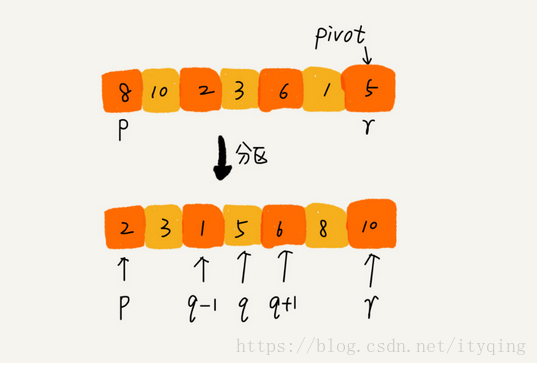


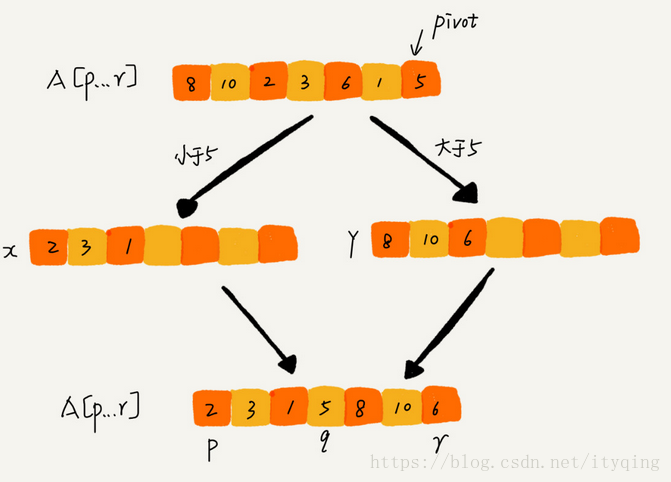
如果要排序一个数组，我们先把数组从中间分为两部分，然后对前后两部分分别排序，再将排好序的两部分合并在一起，这样整个数组就都有序了。

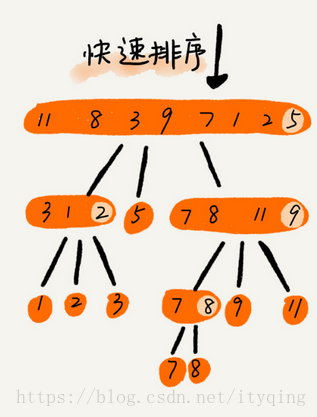
**快速排序**

快速排序的思想：如果要排序数组中下标从p到r之间的一组数据，我们选择p到r之间的任意一个数据作为pivot（分区点）。我们遍历p到r之间的数据，将小于pivot的放到左边，将大于pivot的放到右边,将pivot放到中间。经过这一步骤之后,数组p到r之间的数据就被分成了三个部分,前面p到q-1之间都是小于pivot的,中间是pivot,后面的q+1到r之间是大于pivot的。

快排利用的是分而治之的思想







### **线性排序:**

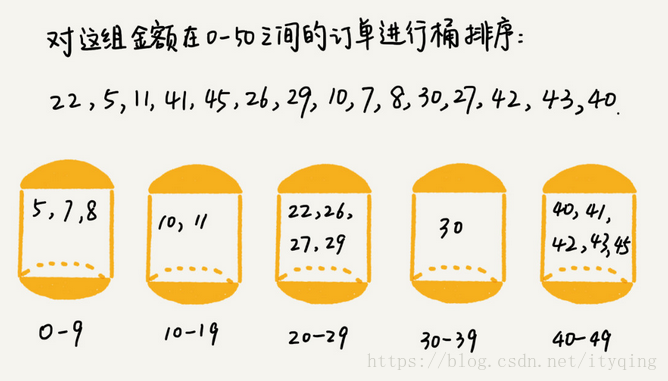
特点:非基于比较的排序算法

**桶排序**

桶排序核心思想是将要排序的数据分到几个有序的桶里，每个桶里的数据再单独进行排序。桶内排完序之后，再把每个桶里的数据按照顺序依次取出，组成的序列就是有序的了。

对排序的数据要求苛刻：

1. 要排序的数据需要很容易就能分为m个桶，并且，桶与桶之间有这天然的大小顺序
2. 数据在各个桶之间的分布是比较均匀的。
3. 桶排序比较适合用在外部排序中。所谓的外部排序就是数据存储在外部磁盘中，数据量比较大，内存有限，无法将数据全部加载到内从中。



**计数排序**

计数排序只能用在数据范围不大的场景中，如果范围k比要排序的数据n大很多，就不适合计数排序了。

计数排序只能给非负数整数排序，如果要排序的数据是其他类型的，要将其在不改变相对大小的情况下，转为非负整数。