1.冒泡排序

> 冒泡排序是一种很简单的排序算法，也是大多数人所学的第一种排序算法，其基本思想是“比较相邻元素，将大的元素放后边，小的放前边，每一轮比较，总有一个最大元素被‘冒’到数组最后边。”N轮比较下来，就完成了排序。

> 时间复杂度：O(n2)
>
> 空间复杂度：原址排序
>
> 是否稳定：是
>
> 应用场景：优化后的冒泡排序可用于当数据已经基本有序，且数据量较小时。

2.选择排序

> 选择排序也是一种简单直观的排序算法，它的基本思想也很简单直观：每次总是选择未排序序列中最小（或最大）的元素，放置在已排序序列最后或最前即可。

> 时间复杂度：O(n2)
>
> 空间复杂度：原址排序
>
> 是否稳定：否
>
> 应用场景：当数据规模较小时，选择排序性能较好

3.插入排序

> 插入排序也是一种基本的排序算法，它的基本思想类似于我们抓扑克牌。开始时，我们的左手为空，然后我们每次从牌堆中拿走一张牌并插入到正确的位置；为了找到每一张牌的正确位置，需要将它从右到左与每一张牌比较，直到找到小于它的那张牌，保证左手的牌随时都是有序的，同样N轮插入操作下来，数组有序。

> 时间复杂度：O(n2)
>
> 空间复杂度：原址排序
>
> 是否稳定：是
>
> 应用场景：若数组基本有序且数据规模较小时，选用插入排序较好.

4.希尔排序

> 希尔排序是对插入排序的一种改进，其改进思想来源于“序列越基本有序，插入排序效率越高”；该算法是第一批突破O(n2)时间复杂度的排序算法。基本思想：先将整个待排记录序列分割成若干列，即若干子序列，分别进行直接插入排序，待整个序列中的记录“基本有序”时，再对全体记录进行一次直接插入排序。

> 时间复杂度：小于O(n4/3)
>
> 空间复杂度：原址排序
>
> 是否稳定：否
>
> 应用场景：数据量较小且基本有序时

5.快速排序

> 快速排序算法每次选择一个基准元素，将小于基准元素的数放到左边，大于基准元素的数放到右边，递归的重复这个划分过程，直到需要划分的数组中只有一个元素，结束递归。

> 时间复杂度：O(nlogn)
>
> 空间复杂度：原址排序
>
> 是否稳定：否
>
> 应用场景：快速排序适合处理大量数据排序时的场景

6.堆排序

> 堆排序利用了二叉堆的性质，维护一个数组，数组可被看成一个近似的完全二叉树，堆顶总是储存数组中最大（或最小）的元素；每次将堆中最大元素取出，与数组最后一位数交换，堆大小减一，这个操作破坏了堆的性质（堆顶不一定为最大元素），于是维护堆的性质，下沉堆顶元素，反复操作后即可得到有序数组。

> 时间复杂度：O(nlog(n))
>
> 空间复杂度：原址排序
>
> 是否稳定：否
>
> 应用场景：堆排序适合处理数据量大的情况，数据呈流式输入时用堆排序也很方便

7.归并排序

> 归并排序在结构上是递归的，归并排序每一次递归将原数组均分为规模较小的两个部分，直到无法再分为止，此时每一个部分只有一个元素，那么自然是有序的，于是递归开始自下往上进行合并； 合并时，新建一个数组，长度等于两个子数组长度之和，依次比较两个子数组中的元素，每次将较小的元素放进新数组即可； 对于归并排序，每一层递归将原数组均分成了两部分，于是递归树一共logn+1logn+1 层，每一层最多进行n次比较，所以时间复杂度为 nlognnlogn 。

> 时间复杂度：O(nlogn)
>
> 空间复杂度：O(n)
>
> 是否稳定：是
>
> 应用场景：数据量较大且要求排序稳定时

8.计数排序

> 本篇博客以上排序算法均属于非线性时间排序算法，现在开始介绍三种线性时间排序算法。由于以上排序均依赖于元素之间的比较，任何比较排序在最坏情况下都需要O(nlogn)O(nlogn) 的时间复杂度，接下来的三种排序算法均可打破这个限制。 限制：计数排序假设n个输入元素中的每一个元素都是在0到k区间内的一个整数，只可处理非负数；计数排序的基本思想：对每一个元素x，确定小于x的元素个数，利用这一信息，就可以直接把x放到输出排序数组的正确位置上了。

> 时间复杂度：O(n)
>
> 空间复杂度：O(n)+O(N)
>
> 是否稳定：是
>
> 应用场景：计数排序虽然时间复杂度较低，但需要满足的条件较多，如果能满足限制条件与空间需求，计数排序自然很快

9.基数排序

> 基数排序是基于计数排序的，基数排序着眼于输入序列的每一位数，每一轮排序都是根据序列中的某一位数进行排序，从低位到高位各进行一次这种排序思想，最终序列便是有序的。由于输入序列每一位数都是有限的，比如十进制序列，每位数都是0~9的数字，于是可以选择计数排序对序列某一位数进行排序。同样，基数排序也不能处理非负数。

> 时间复杂度：O(n)
>
> 空间复杂度：O(n)
>
> 是否稳定：是

> 应用场景：同计数排序

10.桶排序

> 桶排序假设输入数据服从均匀分布，防止所有元素集中在少数几个桶中，平均情况下它的时间代价为O(n)O(n) 。基本思想： 桶排序将[0,1)区间划分为n个相同大小的子区间，也就是桶，然后将n个输入数据分别放到各个桶中，为了得到输出结果，我们队每个桶中的数进行插入排序，最后遍历所有桶按照次序输出数据即可。

> 时间复杂度：O(n)
>
> 空间复杂度：O(n)
>
> 是否稳定：是
>
> 应用场景：如果满足桶排序的假设条件，那么桶排序的速度是非常快的

性能对比小结：

> 1. 传统简单排序确实当数据量很小的时候也表现不错，但当数据量增大，其耗时也增大十分明显；
> 2. 冒泡，插入，选择三种排序中，当数据量很大时，选择排序性能会更好；
> 3. 堆排，希尔，归并，快排几种排序算法也表现不错，源于其时间复杂度达到了O(nlogn)O(nlogn)；
> 4. 随机快速排序性能确实表现十分亮眼，甚至有时比基数排序和桶排序还好，这可能也是快排如此流行的原因；
> 5. 线性排序中计数排序表现最好，但他们的限制也比较明显，只能处理范围内的正整数。