# 算法思想篇（排序-选择&冒泡&快排）

1. 选择排序

选择排序的思路和插入排序非常相似，也分已排序和未排序区间。但选择排序每次会从未排序区间中找到最小的元素，将其放到已排序区间的末尾。但是不像插入排序会移动数组 选择排序会每次进行交换，如以下例子：

4 5 6 3 2 1

第一次：1 5 6 3 2 4

第二次：1 2 6 3 5 4

第三次：1 2 3 6 5 4

第四次：1 2 3 4 5 6

第五次：1 2 3 4 5 6

第六次：1 2 3 4 5 6

1. 时间复杂度：O(N^2)
2. 空间复杂度:O(n)
3. 交换次数
4. 稳定性:不稳定
5. 冒泡排序

核心思路：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互换。一次冒泡会让至少一个元素移动到它应该在的位置，重复n次，就完成了n个数据的排序工作。

举例说明：4 5 6 3 2 1,从小到大排序。

1 2 3 4 5 6进行排序：什么样的情况下不做任何交换了呢，那就是所有的数都在它应该在的位置；O(n)

第一次冒泡的结果:4 5 6 3 2 1->4 5 3 6 2 1 - > 4 5 3 2 6 1 -> 4 5 3 2 1 6,哪个元素的位置确定了，6

第二次冒泡的结果:4 5 3 2 1 6->4 3 5 2 1 6 -> 4 3 2 5 1 6 -> 4 3 2 1 5 6

第三次冒泡的结果:4 3 2 1 5 6->3 4 2 1 5 6 -> 3 2 4 1 5 6 -> 3 2 1 4 5 6

第四次冒泡的结果:3 2 1 4 5 6->2 3 1 4 5 6 -> 2 1 3 4 5 6

第五次冒泡的结果:2 1 3 4 5 6->1 2 3 4 5 6

1. 时间复杂度:O(n^2)
2. 空间复杂度:O(n)
3. 交换次数: O(n^2)
4. 稳定性:稳定

45 28 80 90 50 16 100 10

基准数：一般就是取要排序序列的第一个。

第一次排序基准数：45

从后面往前找到比基准数小的数进行对换：

10 28 80 90 50 16 100 45

从前面往后面找比基准数大的进行对换：

10 28 45 90 50 16 100 80

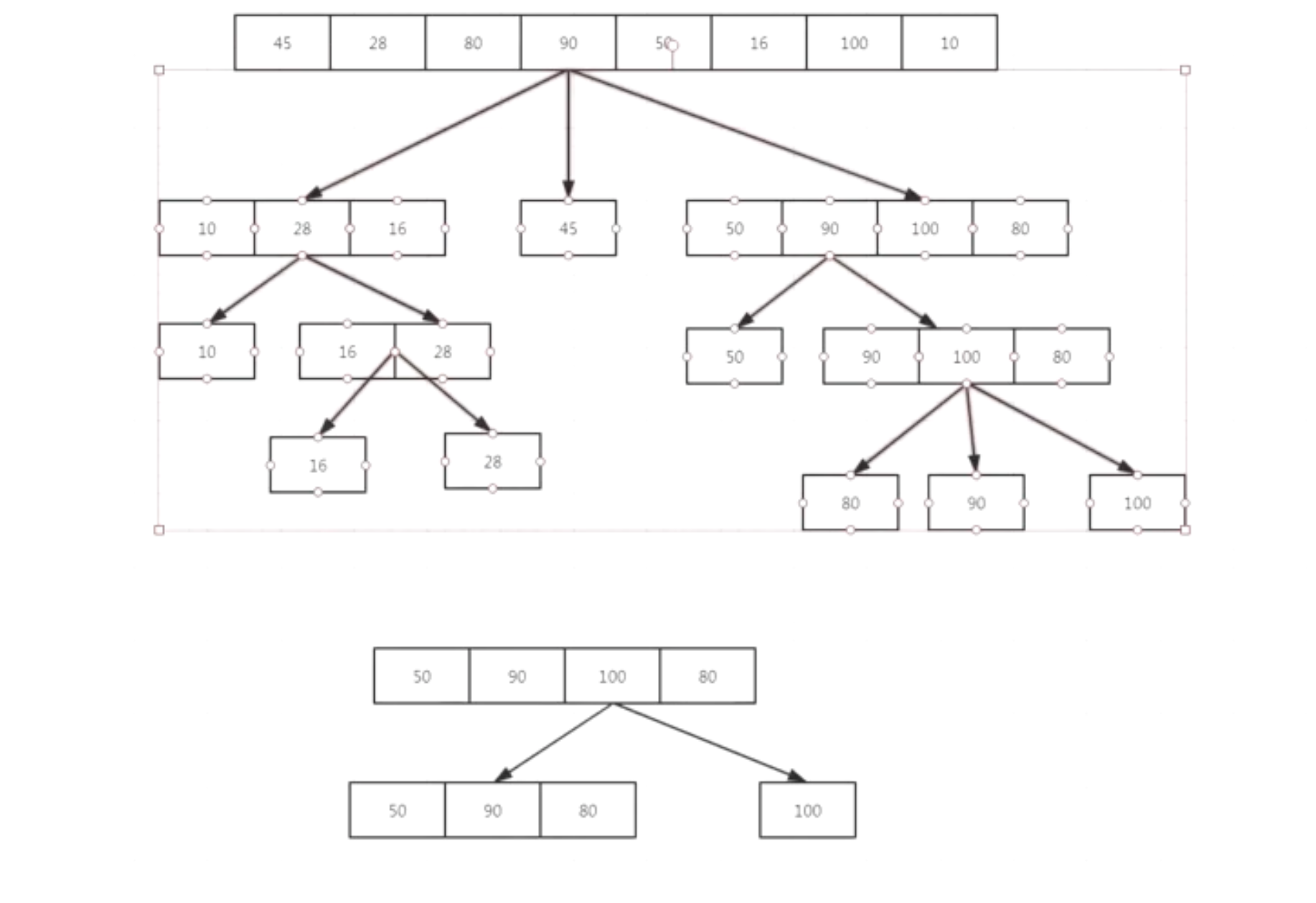
10 28 16 90 50 45 100 80

10 28 16 45 50 90 100 80

以基准数分为3部分，左边的比之小，右边比之大：

{10 28 16} 45 {50 90 100 80}

到此第一次以45位基准数的排序完成。



1.时间复杂度:nlogn 最坏的情况就是O(n^2)

2.空间复杂度:O(n)

3.稳定性：不稳定

4.快排和归并的对比：

（1）归并排序的处理过程是由下到上的，先处理子问题，然后再合并。

（2）快排其实就是从上到下，先分区，在处理子问题，不用合并。

其优化就是优化基准数,提供一个取三个数中间的思路。

这么多种排序算法我们究竟应该怎么选择呢？:

1.分析场景：稳定还是不稳定

2.数据量：数据量小的时候选什么？比如就50个数，优先选插入（5000\*5000=25000000）

3.分析空间：

综上所述，没有一个固定的排序算法，都是要根据情况分析的。但是如果你不会分析的情况下 选择归并或者快排。

C++ qsort：快排+插入排序

jdk里面有arrays.sort:一种是基础类型，int double 用的快排。对象排序，用的是归并+timeSort

|  |  |  |  |
| --- | --- | --- | --- |
| **排序名称** | **时间复杂度** | **是否稳定** | **额外空间开销** |
| 插入排序 | O(n^2) | 稳定 | O(1) |
| 冒泡排序 | O(n^2) | 稳定 | O(1) |
| 选择排序 | O(n^2) | 不稳定 | O(1) |
| 希尔排序 | O(n^2) | 不稳定 | O(1) |
| 归并排序 | O(nlogn) | 稳定 | O(n) |
| 快速排序 | O(nlogn) | 不稳定 | O(1) |