# 排序算法比较

代码行数：398行

算法思想：每次生成一个样本用不同的排序测试并记录所用时间，一共十个样本，最后输出结果。

主要/核心函数分析：堆排序：将带排序的序列构造成一个大顶堆，根据大顶堆的性质，当前堆的根节点（堆顶）就是序列中最大的元素，将堆顶元素和最后一个元素交换，然后将剩下的节点重新构造成一个大顶堆。归并排序：将一个数组拆分为两个，从中间点拆开，通过递归操作来实现一层一层拆分，从左右数组中选择小的元素放入到临时空间，并移动下标到下一位置，重复前面步骤直到某一下标达到尾部，将另一序列剩下的所有元素依次放入临时空间，将临时空间的数据依次放入原数据数组。快速排序：通过一趟排序将待排记录分隔成独立的两部分，其中一部分记录的关键字均比另一部分的关键字小，则可分别对这两部分记录继续进行排序，以达到整个序列有序。基数排序：将所有待比较数值统一为同样的位数长度，数位较短的数前边补零。然后，从最低位开始，依次进行一次排序，这样从最低位排序一直到最高位排序完成后，就变成一个有序数列。选择排序：第一次从arr[0到]arr[n-1]中选取最小值，与arr[0]交换，第二次从arr[1]到arr[n-1]中选取最小值，与arr[1]交换，第三次从arr[2]到arr[n-1]中选取最小值，与arr[2]交换，…，第i次从arr[i-1]arr[n-1]中选取最小值，与arr[i-1]交换，…, 第n-1次从arr[n-2]~arr[n-1]中选取最小值，与arr[n-2]交换，总共通过n-1次，得到一个按排序码从小到大排列的有序序列。希尔排序：把记录按下标的一定增量分组，对每组使用直接插入排序算法排序；随着增量主键递减，每组包含的关键词也来越多，当增量减至1时，整个文件恰被分成一组，算法终止。插入排序：构建有序序列，对于未排序数据，在已排序序列中从后向前扫描，找到相应位置并插入。插入排序在实现上，在从后向前扫描过程中，需要反复把已排序元素逐步向后挪位，为最新元素提供插入空间。

测试数据：系统随机生成的10个样本，每个样本有50000个随机整数

运行结果：

排序算法/时间

插入排序：922 859 875 875 875 890 859 860 859 875

希尔排序：15 0 0 0 15 16 16 15 0 0

冒泡排序：4703 4610 4625 4609 4594 4640 4625 4625 4672 4609

快速排序：0 0 15 0 0 0 0 0 0 0

选择排序：1360 1375 1375 1360 1375 1375 1375 1375 1390 1375

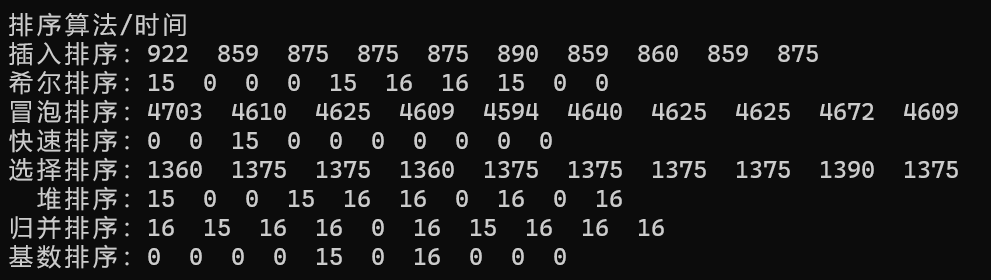
堆排序：15 0 0 15 16 16 0 16 0 16

归并排序：16 15 16 16 0 16 15 16 16 16

基数排序：0 0 0 0 15 0 16 0 0 0

时间复杂度：冒泡，插入排序：O（n\*n）。希尔，希尔，快速，归并，堆排序：O（n\*logn）。基数排序O（k\*n）。

结果截屏：



心得体会：学会了八大排序算法的使用，更直观体验了算法的快慢，对于稳定性的体验自己也用其他样本做过测试。学会如何获取运行时间以及生成随机数。