

HW3

林宸昊 PB20000034

1. 插入排序，归并排序和计数排序稳定，快速排序和堆排序不稳定。

为使排序算法稳定，可以在接受一组数据时按照接受顺序为数据编号，例如要从小到大排列，则将小于重新定义为

$$\circ (i < j) \parallel (i == j) \&\& (i.index < j.index)$$

由于编号全不相同，故不会出现相同序号再次导致排序不稳定。这样做会使空间开销加倍，但时间复杂度不变（微观上时间会略有增加）。

2. 由于总位数确定，可以先以位数为基准将数据分组，位数少的一定小于位数多的；然后对于每一组采用基数排序，最后根据位数从小到大依序输出各组排序好的数据即可完成整体排序。

算法的时间复杂度分析如下

- 首先将数据按位数分组。假设所有数据位数分别为

$$b_1, b_2, \dots, b_n$$

得到一个数据的位数 b_i 所需时间复杂度为 $\Theta(b_i)$ ，即遍历该数据每一位。

又由条件显然有

$$b_1 + b_2 + \dots + b_n = n, \Theta(b_1) + \Theta(b_2) + \dots + \Theta(b_n) = \Theta(n)$$

即得到所有数据位数需时间 $\Theta(n)$ 。

- 然后将同一组拥有同样位数的数据进行基数排序。对于位数为 i 的组中的 x_i 个数据，利用基数排序所需时间复杂度为 $\Theta(i * x_i)$ 。进行加总有：

$$\sum \Theta(i * x_i) = \Theta(n)$$

即最后总的时间复杂度为 $\Theta(n) + \Theta(n) = O(n)$ 。

3. 根据题意，只需要考虑 $i < n/2$ 的情况，当 $i \geq n/2$ 时，使用原SELECT算法即可。假定输入 n 个数据， $m = \lfloor n/2 \rfloor$

- 将数据平均分为两组：

$$\begin{aligned} & a[1], a[2], a[3], \dots, a[m] \\ & b[m+1], b[m+2], b[m+3], \dots, b[2m] \end{aligned}$$

然后将每一对对应的进行比较：

■

$$(a[1], b[m + 1]), \dots, (a[m], b[2m])$$

可以得到一个由全部数对中较小元素组成的序列：

$$c[1], c[2], c[3], \dots, c[m]$$

对这一序列递归使用一次该算法得到其中第 i 小的元素以及所有比这个元素小的元素。将这些元素在第二步中所对应的数对中的另一元素与这些元素放在一起，并使用一次SELECT。此时SELECT对象的数据规模不超过 $2i$ ，所需次数为 $T(2i)$ 。而 i 为常数，计算次数时该项退化为 $T(1)$ 。

同样，从 $\lfloor n/2 \rfloor$ 二分到达 i 这一规模所需次数为 $\lg \lfloor n/(2i) \rfloor$ ，由于 i 为常数，退化为 $\lg n$ 。故最终得到的比较次数为每一次组间比较次数之和加上 $O(\lg n)$ 。

$$\begin{aligned} C_i(n) &= \lfloor n/2 \rfloor + \lfloor n/4 \rfloor + \dots + O(\lg n) \\ &< n + O(\lg n) \end{aligned}$$

满足题意。