

# 实验报告

比较Insertion sort、Shellsort、Quicksort、Merge sort、Radix sort对32位无符号整数的排序结果，并探究不同数据规模（ $10^1 \sim 10^8$ ）下不同方法排序时间的差异。

## 实验环境

CPU: Intel Core i5-7200U

内存: 8GB

操作系统: Windows 10 64位教育版

编程语言: C++

编译环境: MSVC 2017 64-bit

## 算法分析

### Insertion Sort

插入排序，其时间复杂度 $T(n) = O(n^2)$

### Shellsort

希尔排序，其时间复杂度 $T(n) = \Theta(n \lg n)$

### Quicksort

快速排序，其时间复杂度 $T(n) = O(n^2)$ ，在本实验中由于数组的均匀随机性可认为 $T(n) = \Theta(n \lg n)$

### Merge sort

归并排序，其时间复杂度 $T(n) = \Theta(n \lg n)$

### Radix sort

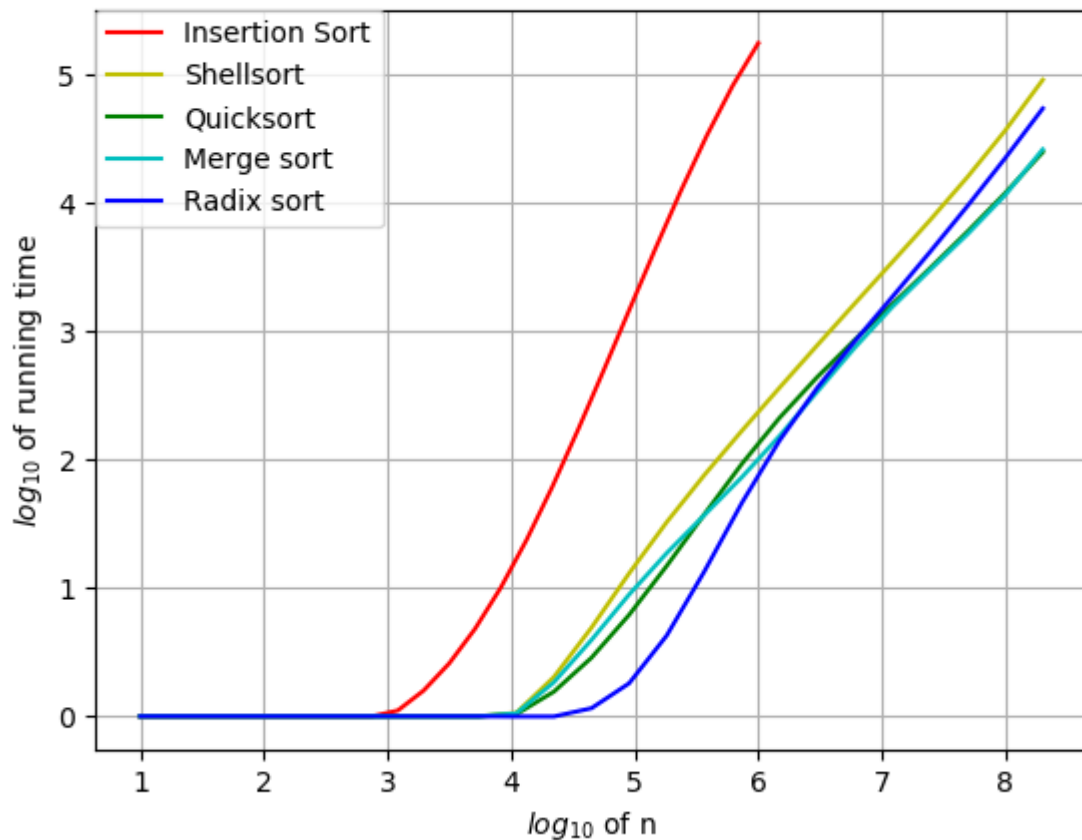
基数排序，其时间复杂度 $T(n) = \Theta(d(n + k))$ ，其中 $d$ 为数字的位数， $k$ 为每个数位可能的取值，在本实验中可近似认为 $T(n) = \Theta(n)$

## 结果分析

由于实验环境内存大小的限制，取 $n = 10^1 \sim 10^8$ ，分别使用5种算法进行实验，每组实验均进行5次取运行时间的平均值，并记录其运行时间和计算结果，运行时间如下表（单位为ms）：

$n$	Insertion Sort	Shellsort	Quicksort	Merge sort	Radix sort
$10^1$	1	0	0	0	1
$10^2$	1	0	0	0	0
$10^3$	0	0	0	0	0
$10^4$	14	1	1	1	0
$10^5$	1852	15	7	10	2
$10^6$	176758	236	135	100	77
$10^7$	严重超时	2828	1367	1280	1490
$10^8$		37108	12056	11629	22611
$2 \times 10^8$		91464	24963	26280	54710

由数据表格可以初步看出， $\Theta(n^2)$ 的插入排序算法在运行速度上相比其他4种算法有明显差距。为了更加明确不同输入规模下两种算法的差异，对数据进行简单处理后画出了如下的曲线图：



分析上图可得，当 $n < 10^3$ 时，由于数据规模较小，5种算法用时均很少，但当 $n > 10^3$ 时， $\Theta(n^2)$ 的插入排序算法的运行时间迅速超过其他4种算法，到 $n = 10^7$ 时已经严重超时。其他4种 $O(n \lg n)$ 算法中，希尔排序的速度最慢，快速排序和归并排序的时间基本相等。当 $n < 10^7$ 时，基数排序由于其线性时间优势，运行速度最快；当 $n > 10^7$ 时，基数排序的 $T(n)$ 受到常数项因素的影响而增加，快速排序和归并排序用时最短。在空间复杂度上，归并排序和基数排

序都需要与数组相同大小的辅助数组，在 $n > 10^8$ 后对内存占用有一定限制。

综上，在对随机均匀数组进行排序时，若数据规模 $n < 10^7$ ，推荐使用基数排序算法，若 $n > 10^7$ ，应该考虑使用快速排序算法。