

Week 5

软73 沈冠霖 2017013569

March 26, 2019

1 T1

根据书上8.4, 想要用桶排序把一组均匀分布的数以 $\theta(n)$ 的复杂度排序, 则需要将其分成 n 部分, 其中每个数出现在每一部分的概率都是 $\frac{1}{n}$ 。因此可以把圆分成 n 组, 第 i 组中的点满足 $\frac{\sqrt{i-1}}{\sqrt{n}} \leq \sqrt{x^2 + y^2} \leq \frac{\sqrt{i}}{\sqrt{n}}$, 这样每个部分的面积都是 $\frac{1}{n}$, 任意一个点出现在每部分概率也是 $\frac{1}{n}$

2 T2

假设长度为 i 的钢条价值为 $v[i]$, 切割一次成本为 c , 长度为 n , 设长度为 i 钢条最大利润为 $r[i]$, 则仍然考虑长为 i 的钢条, 最右一次切割所在的位置。
状态转移方程为: $r[i] = \max(v[i], r[i-1] + v[1] - c, r[i-2] + v[2] - c, \dots, r[1] + v[i-1] - c)$

3 T3

测试环境 CPU:Inter Core i5-6300HQ,2.3GHZ

内存: 12G

环境: VS2017, release模式 比较不同数据规模下五种排序算法的时间的结果, 具体数据见Table 1 与下方折线图。

结果分析 运行结果全部正确。运行时间分析如下: 首先, 在数据范围较小 ≤ 1000 的时候, 五种算法都几乎一样快, 因为在 n 很小的时候, $O(n^2)$, $O(n \lg n)$, $O(n)$ 差别不大。

其次, 在数据规模满足 $1000 \leq n$ 的时候, $O(n^2)$ 的插入排序和希尔排序时间增加很快, 乃至在超过100万的时候排序时间过长。而因为我选择希尔排序第一趟间隔为7, 希尔排序时间始终是插入排序的 $\frac{1}{8} - \frac{1}{7}$ 之间, 也符合预期。而此时, 归并, 基数排序时间大致相当, 且都是接近线性增长。而 $O(n \lg n)$ 的快速排序则略慢, 而且在数据范围过大的时候排序明显慢。因为基数排序是 $\theta(n)$ 的算法, 比较快很正常。而我猜测, 归并排序快是因为其每步操作较为简单, 常数较小。而且其每次归并可能有达 $\frac{1}{3}$ 的元素不用归并, 它们大小不足

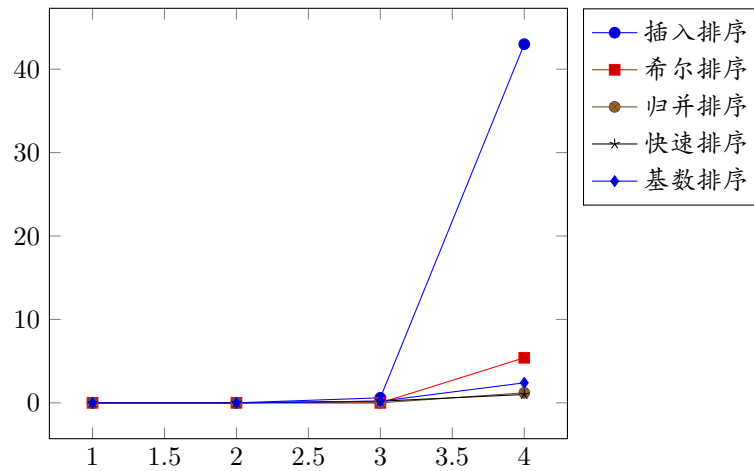
一个归并段。

最后，虽然基数排序和归并排序较快，其至少需要大小为 n 的额外空间，20亿规模的数据已经达到内存空间的极限了。

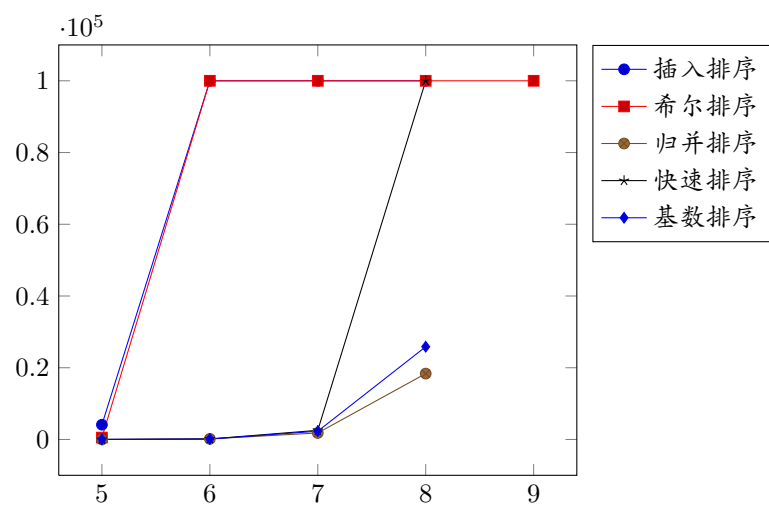
Table 1: 不同数据规模下排序的时间

测量序号	1	2	3	4	5	6	7	8	9
数据范围	10	100	1000	10000	10^5	10^6	10^7	10^8	$2 * 10^8$
插入排序时间 (ms)	0	0	0.6	43	4113.4	太大	太大	太大	太大
希尔排序时间 (ms)	0	0	0	5.4	524.6	太大	太大	太大	太大
归并排序时间 (ms)	0	0	0	1.2	15.4	175.8	1813.6	18366	39362
快速排序时间 (ms)	0	0	0.2	1	11.4	147.8	2553.6	134330	太大
基数排序时间 (ms)	0	0	0.2	2.4	18.2	113.6	2304.6	25861.2	53505

注：除了 10^8 数量级只测试了2组外，每组数据都是运行5次后取的平均值



注：横轴代表 $\log_{10} n$, 对于超过100s的排序，其时间显示为100s



注：横轴代表 $\log_{10} n$, 对于超过100s的排序，其时间显示为100s