

9. 词典

(f) 基数排序

邓俊辉

deng@tsinghua.edu.cn

基数排序：词典序

❖ 有时，关键码由多个域组成： k_t, k_{t-1}, \dots, k_1

❖ 若将各域视作字母，则关键码即单词

❖ 于是，可按照词典方式排序 `lexicographic order`

❖ $\spadesuit A > \spadesuit K > \spadesuit Q > \spadesuit J > \spadesuit 10 > \dots > \spadesuit 2$

$\heartsuit A > \heartsuit K > \heartsuit Q > \heartsuit J > \heartsuit 10 > \dots > \heartsuit 2$

$\diamondsuit A > \diamondsuit K > \diamondsuit Q > \diamondsuit J > \diamondsuit 10 > \dots > \diamondsuit 2$

$\clubsuit A > \clubsuit K > \clubsuit Q > \clubsuit J > \clubsuit 10 > \dots > \clubsuit 2$

基数排序：算法

❖ 低位优先：自 k_1 到 k_t ，依次以各域为序，做一趟桶排序

❖	3280	5112	5112	4320	6441
	4320	7214	7214	3280	5276
	4698	5276	4320	6441	4320
	5112	3280	6441	5112	7214
	5276	4320	5276	7214	4698
	6441	6441	3280	5276	3280
	7214	4698	4698	4698	5112

基数排序：正确性

❖ 以上算法的正确性何以见得？**数学归纳**

❖ 归纳假设：在经过算法的**前 i 趟**之后，所有词条关于**最低的 i 位**有序——**第1趟**易见成立

❖ 假设**前 $i - 1$ 趟**均成立，现考查**第 i 趟**排序之后的时刻

无非两种情况：1) 凡第 i 位**不同**的词条：即便此前**曾是**逆序，现在亦必已**转为**有序

2) 凡第 i 位**相同**的词条：得益于桶排序的**稳定**性，必**保持**原有次序

❖ 由此也可看出，只要实现**得法**，基数排序同样**稳定**

基数排序：性能

❖ 运行时间

= 各趟桶排序所需时间之和

= $(n + 2m_1) + (n + 2m_2) + \dots + (n + 2m_t)$ // m_k 为各域的取值范围

= $O(t \times (n + m))$ // $m = \max\{m_1, \dots, m_t\}$

整数排序

- ❖ 对于 $[0, n^d)$ 内的任意 n 个整数，如何高效排序？ // 常数 $d > 1$
- ❖ 理解： $1/d = \log(n) / \log(n^d)$ // 常对数密度，实际应用中不难满足
- ❖ 预处理：将所有关键码转换为 n 进制形式 $x = (x_d, \dots, x_2, x_1)$
- ❖ 于是，原问题转化为 d 个域的基数排序问题，可套用前述算法
- ❖ 排序时间 $= d(n + n) = O(n)$ // “突破”了此前确定的下界！
- ❖ 原因：1) 整数取值范围有限制；2) 不再是基于比较的计算模式
- ❖ 预处理的用时尚未计入，它本身需要多少时间？回忆一下，此前的相关内容...