

11D 桶排序/计数排序

#数据结构邓神

算法的性能取决于待排序的规模和取值范围

 $[0, M)$

时间复杂度

$$O(n + M) == O(\max(n, m))$$

待排序取值范围越是有限，效率越高！

例如，清华大学在校本科生总数约为

$$n = 3,200 \times 4 = 13,000$$

若需按照**誕生日 (Date Of Birth)** 对他们排序, 则大致有

$$M = 365 \times 4 = 1,500$$

而若按照生日 (Birthday) 排序, 则有

M = 366

无论如何，都存在大量的重复

例子 对于大量英文字母进行排序



accum[]	0	2	2	5	7	10	11	11	11	13	14	15	18	20	21	21	23	26	26	26	26	26	27	28	30	32
count[]	0	2	0	3	2	3	1	0	0	2	1	1	3	2	1	0	2	3	0	0	0	0	1	1	2	2
key/value	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

[illegible]

这类问题，通常都满足

$$M = 26 \ll n$$

故存在大量的重复

蓝色折线，对应于输入的随机序列

红色折线，对应于输出的排序序列

后者必然单调

计算过程 | 原理

运用散列表，各个元素的取值无非26种可能。

建立散列表

散列函数: $\text{Value} - 'A'$