

算法设计与分析作业二

作者：吴润泽 学号：181860109

Email: 181860109@smail.nju.edu.cn

2020 年 3 月 6 日

目录

PART I	2
problem 6.8	2
problem 6.9	4
problem 6.10	5
problem 6.13	5
problem 6.15	5
 PART II	 6
problem 7.1	6
problem 7.2	7
problem 7.3	7
problem 7.4	7
problem 7.6	7

PART I

problem 6.8

算法分析 假定 n 总是 k 的倍数, 且 n 和 k 都是 2 的幂。

利用快排的思想, 将数组从中间划分为两段 $A[0 \cdots n/2]$, $A[n/2+1 \cdots n]$, 且左段元素小于右段元素。

对于子序列继续递归划分, 得到 $A[0 \cdots n/2^m]$, $A[n/2^m+1 \cdots n/2^{m-1}] \cdots A[n - n/2^m + 1 \cdots n]$, 当 $2^m = k \rightarrow m = \log k$ 时, 划分完成。

因此寻找中位数划分的函数时间复杂度应为 $O(n)$, 划分函数的递归方程为 $W(n) = 2W(n/2) + O(n)$, 划分左右子段 $\log k$ 次, 方能使得总的时间复杂度达到 $O(n \log k)$ 。

具体算法实现请见下页

算法时间复杂度 对于 `findk_pos`, 每次递归代价为 $O(n)$, 每次子问题缩小为原来一半的规模, 且子问题只有一个, 可列出递归方程为 $T(n) = T(n/2) + O(n)$, 由主定理可以得出 $T(n) = O(n)$ 。

对于 `k_sorted`, 每次递归代价为 $O(n)$, 每次子问题缩小为原来一半, 而需要划分左右两序列, 子问题为两个, 可列出递归方程为 $W(n) = 2W(n/2) + O(n)$, 注意结束条件为递归调用了 $\log k$ 层, 每层代价均为 $O(n)$, 因此时间复杂度为 $O(n \log k)$ 。

算法 1 k-sorted 算法

输入: 待划分序列 $A[1 \cdots n]$, 划分段数 k **输出:** 划分后的的序列 A

```

1: function FINDK_POS( $A, k\_pos, begin, end$ ) \\ 返回该段数组第  $k$  小
2: /* 利用快排思想, 选定一个 key, 将大于 key 的元素放在其右边, 小于
   key 放于左边。判断 key 插入的位置是否为  $k$ , 如果是则函数返回, 如
   果插入位置大于  $k$  说明第  $k$  小位于左子序列对左边递归寻找, 否则对右
   子序列递归寻找。*/
3:    $split \leftarrow begin, key \leftarrow A[begin]$ 
4:   for  $i \leftarrow begin + 1$  to  $end$  do
5:      $A[i] \leq key ? swap(A[++split], A[i])$ 
6:   end for
7:    $split > k\_pos ? \text{return } findk\_pos(A, k\_pos, begin, split - 1)$ 
8:    $split < k\_pos ? \text{return } findk\_pos(A, k\_pos, split + 1, end)$ 
9:   return  $split$ 
10: end function
11: function K_SORTED( $A, begin, end, k, count = 1$ )
12: /* count 记录当前的段数, 每次调用 findk_pos,  $A$  被分为  $[begin, mid]$ 
   和  $[mid+1, end]$  两段, 段数变为原来两倍, 且左段元素小于右段, 调用
   层数达到  $\log k$  层算法结束, 否则继续划分左右子序列 */
13:    $mid \leftarrow (end - begin)/2 + begin, count \leftarrow count * 2$ 
14:    $findk\_pos(A, mid, begin, end)$ 
15:   if  $count == k$  then
16:     划分  $k$  段, 算法结束
17:     return  $A$ 
18:   end if
19:    $k\_sorted(A, begin, mid, k, count)$ 
20:    $k\_sorted(A, mid + 1, end, k, count)$ 
21: end function

```

problem 6.9

算法 2 对三个数进行排序**输入:** 含有三个各不相同的整数的序列 $A = \{a, b, c\}$ **输出:** 从小到大排好序的序列 A

problem 6.10

problem 6.13

problem 6.15

PART II

problem 7.1

problem 7.2

problem 7.3

problem 7.4

problem 7.6