

《算法设计与分析》分治作业第二题答案

姓名: XXX

学号: XXXXXXXXX

题目2: 设 $X[0:n-1]$ 和 $Y[0:n-1]$ 为两个数组, 每个数组中含有 n 个已排好序的数。试设计一个 $O(\log n)$ 时间的分治算法, 找出 X 和 Y 的 $2n$ 个数的中位数, 并证明算法的时间复杂性为 $O(\log n)$ 。

答: 包括算法思路, 伪代码和时间复杂度分析

算法思路: 简单来说, 就是比较两个区间的中位数, 如果第一个区间的中位数比第二个大, 那么就把第一个区间的范围缩小至它的前半段, 把第二个区间缩小至它的后半段, 然后重复上述过程。具体思路如下所示:

对于两个数组 X, Y , 我们可以从他们中分别选取出一个中位数, 并将两个数组的左右边界称之为 $xLeft, xRight, yLeft, yRight$ 。比较两个中位数, 元素个数分为奇数个数和偶数个数两种情况:

1、当两个数组中的元素个数都为偶数时:

(1) 如果 X 数组的中位数大于 Y 数组的中位数, 则 X 数组被切割为 $X[xLeft, x+1]$ (x 表示 X 的中位数的下标), Y 数组被切割为 $Y[y, yRight]$ (y 表示 Y 的中位数下标);

(2) 如果 X 数组的中位数小于 Y 数组的中位数, 则 X 数组被切割为 $X[x, xRight]$, Y 数组被切割为 $Y[yLeft, y+1]$ 。

(3) 如果两个数组的中位数相等, 则该中位数即为结果, 递归结束。

2、当两个数组中的元素个数都为奇数时:

(1) 如果 X 数组的中位数大于 Y 数组的中位数, 则 X 数组被切割为 $X[xLeft, x]$, Y 数组被切割为 $Y[y, yRight]$;

(2) 如果 X 数组的中位数小于 Y 数组的中位数, 则 X 数组被切割为 $X[x, xRight]$, Y 数组被切割为 $Y[yLeft, y]$ 。

(3) 如果两个数组的中位数相等, 则该中位数即为结果, 递归结束。

据上所述, 对于原问题 $X[xLeft, xRight], Y[yLeft, yRight]$ 。假设切割后的子问题为 $X[xLeft, x+1], Y[y, yRight]$ 。则求解 $X[xLeft, xRight], Y[yLeft, yRight]$ 问题的中位数, 可归结于求解子问题 $X[xLeft, x+1], Y[y, yRight]$ 的中位数。

该算法可以使两个数组的元素个数始终保持相同个数, 当两个数组的中位数相等时, 直接返回该结果; 或者当切割后得到的子问题的两个数组的长度都为2时, 整个递归结束, 并在此时计算出这四个数的中位数。

时间复杂度:

该算法的递推关系式为:

$$T(n) = T(n/2) + O(1)$$

容易得到该算法的时间复杂度为 $O(\log n)$ 。

伪代码:

Algorithm 1 分治法求两有序数组的中位数

Input: 两个有序数组 X, Y ;

Output: 两个有序数组的中位数;

```
1: function FINDMEDIAN( $doubleX[], intxLeft, intxRight, doubleY[], intyLeft, intyRight$ )
2:   if  $xRight - xLeft = 1$  and  $yRight - yLeft = 1$  then
3:     return 这四个数的中位数
4:   end if
5:    $xMedian \leftarrow (xLeft + xRight)/2$ 
6:    $yMedian \leftarrow (yLeft + yRight)/2$ 
7:    $xResult \leftarrow X$ 数组的中位数
8:    $yResult \leftarrow Y$ 数组的中位数
9:   if  $xResult = yResult$  then
10:    return  $xResult$ ;
11:  end if
12:  if  $X$ 数组元素个数是奇数 then
13:    if  $xResult > yResult$  then
14:       $findMedian(X, xLeft, xMedian, Y, yMedian, yRight)$ 
15:    else  $xResult < yResult$ 
16:       $findMedian(X, xMedian, xRight, Y, yLeft, yMedian)$ 
17:    end if
18:  end if
19:  if  $X$ 数组元素个数是偶数 then
20:    if  $xResult > yResult$  then
21:       $findMedian(X, xLeft, xMedian + 1, Y, yMedian, yRight)$ 
22:    else  $xResult < yResult$ 
23:       $findMedian(X, xMedian, xRight, Y, yLeft, yMedian + 1)$ 
24:    end if
25:  end if
26: end function
```
