Assignment 2 of Algorithm Design and Analysis

L.J.SHOU 寿林钧
No. **201228013229133**

Oct.22 2012

1 Divide and Conquer(2)

QUESTION:

Try to find the median number from two databases, whose sizes are n respectively in O(lgn) time complexity.

Notes: the given databases can only be accessed through quiries.

Pseudo-Code:

Get-Median$(A, B, n, a, b)A, B$ denotes databases;
1: $k = [n/2]$;
2: **if** $n = 1$ **then**
3: **return** $\min(A(a + k), B(b + k))$;
4: **endif**
5: **if** $A(a + k) < B(b + k)$ **then**
6: **return Get-Median**$(A, B, k, a + k, b)$;
7: **else**
8: **return Get-Median**$(A, B, k, a, b + k)$;
9: **endif**

Complexity Analysis:

$T(n) <= T(n/2) + 2$
As a result, we have $T(n) = O(lgn)$

2 Sorting(2)

My test results are shown in the followig table, which are based on three random integer arrays whose sizes are 100,000, 1,000,000, 10,000,000 respectively.

Comparision of different Sorting Methods

No. / **METHODS**	10^5	10^6	10^7
MergeSort	0.05s	0.62s	7.07s
QuickSort	0.03s	0.40s	4.75s
M_QuickSort	0.04s	0.42s	5.04s
MergeSort_Stack	0.06s	0.68s	7.71s
QuickSort_Stack	0.04s	0.50s	5.72s
M_QuickSort_Stack	0.04s	0.48s	5.47s

while if the input integer arrays are already in order, running time VARIES distinctively.

No. / **METHODS.**	10^5
MergeSort	0.04s
QuickSort	59.71s
M_QuickSort	0.02s
MergeSort_Stack	0.04s
QuickSort_Stack	59.71s
M_QuickSort_Stack	0.03s