Data Structures and Algorithms
Pennsylvania State University
Professor Paul Medvedev

September 18, 2015
CMPSC 465

Homework 3 – solution

1. The input is two sorted arrays. Each array has n elements. In total, there are 2n elements. You can assume that all values in the arrays are distinct. A statistician asks you to find the nth order statistic of these values. The nth order statistic is the nth smallest value.

   (a) Give a divide and conquer algorithm to find this value in asymptotic time $T(\log n)$.

   (b) Argue why your algorithm is correct.

   (c) Write down the running time recurrence, including the base case

   (a) Here is a possible solution, given using pseudocode:

```
MEDIAN($A$, $A_{start}$, $A_{end}$, $B$, $B_{start}$, $B_{end}$)
▷ returns the median of $A[A_{start}..A_{end}]$ and $B[B_{start}..B_{end}]$
▷ to find the median of $A$ and $B$, you would call MEDIAN($A$, 1, $A.length$, $B$, 1, $B.length$)
$midA \leftarrow A_{start} + \lfloor \frac{A_{end}-A_{start}}{2} \rfloor$
$midB \leftarrow B_{start} + \lfloor \frac{B_{end}-B_{start}}{2} \rfloor$
if $A_{start} = A_{end}$ ▷ only one element in each array
    then
        return $\min(A[A_{start}], B[B_{start}]$.
if $A[midA] > B[midB]$
    then
        return MEDIAN($A$, $A_{start}$, $midA$, $B$, $B_{start} + \lceil \frac{B_{end}-B_{start}}{2} \rceil$, $B_{end}$).
    else
        return MEDIAN( $A$, $A_{start} + \lceil \frac{A_{end}-A_{start}}{2} \rceil$, $A_{end}$), $B$, $B_{start}$, $midB$).
```

   (b) The basic idea is to compare the two middle elements of each array. Let us say without loss of generality that the A element is larger. Then all the elements in the right half of A are larger than the elements in the left half of A and of B (of which there are at least n). Therefore, those elements in the right half of A cannot be the median. Similarly, the left elements of B are smaller than the right elements of both B and A, meaning that there are at least $n+1$ smaller elements. Hence, those left elements of B cannot be the median either. One has to be careful to consider what can be said about the middle elements themselves. Also, the code takes into account that n can be both even or odd.

   (c) The running time is given by

$$T(n) = T(n/2) + \Theta(1)$$

2. Solutions to Problem 2 are available on Angel via Review Mode.