Examples

Ex 1

a. Write pseudocode for a divide-and-conquer algorithm for finding the position of the largest element in an array of n numbers.

b. What will be your algorithm's output for arrays with several elements of the largest value?

c. Set up and solve a recurrence relation for the number of key comparisons made by your algorithm.

d. How does this algorithm compare with the brute-force algorithm for this problem?

a. Call **Algorithm** $MaxIndex(A[0..n-1])$ where

Algorithm $MaxIndex(A[l..r])$
//Input: A portion of array $A[0..n-1]$ between indices l and r $(l \leq r)$
//Output: The index of the largest element in $A[l..r]$
if $l = r$ **return** l
else $temp1 \leftarrow MaxIndex(A[l..\lfloor(l+r)/2\rfloor])$
    $temp2 \leftarrow MaxIndex(A[\lfloor(l+r)/2\rfloor+1..r])$
    if $A[temp1] \geq A[temp2]$
        return $temp1$
    else return $temp2$

a. Write pseudocode for a divide-and-conquer algorithm for finding values of both the largest and smallest elements in an array of n numbers.

b. Set up and solve (for $n = 2^k$) a recurrence relation for the number of key comparisons made by your algorithm.

c. How does this algorithm compare with the brute-force algorithm for this problem?

a. Call **Algorithm** $MinMax(A[0..n-1],\ minval,\ maxval)$ where

Algorithm $MinMax(A[l..r],\ minval,\ maxval)$
//Finds the values of the smallest and largest elements in a given subarray
//Input: A portion of array $A[0..n-1]$ between indices l and r $(l \le r)$
//Output: The values of the smallest and largest elements in $A[l..r]$
//assigned to $minval$ and $maxval$, respectively
if $r = l$
    $minval \leftarrow A[l]$; $maxval \leftarrow A[l]$
else if $r - l = 1$
        if $A[l] \le A[r]$
            $minval \leftarrow A[l]$; $maxval \leftarrow A[r]$
        else $minval \leftarrow A[r]$; $maxval \leftarrow A[l]$
else $//r - l > 1$
    $MinMax(A[l..\lfloor(l+r)/2\rfloor],\ minval,\ maxval)$
    $MinMax(A[\lfloor(l+r)/2\rfloor + 1..r],\ minval2,\ maxval2)$
    if $minval2 < minval$
        $minval \leftarrow minval2$
    if $maxval2 > maxval$
        $maxval \leftarrow maxval2$

a. Write pseudocode for a divide-and-conquer algorithm for the exponentiation problem of computing a^n where n is a positive integer.

b. Set up and solve a recurrence relation for the number of multiplications made by this algorithm.

a. The following divide-and-conquer algorithm for computing a^n is based on the formula $a^n = a^{\lfloor n/2 \rfloor} a^{\lceil n/2 \rceil}$:

Algorithm $DivConqPower(a, n)$
//Computes a^n by a divide-and-conquer algorithm
//Input: A positive number a and a positive integer n
//Output: The value of a^n
if $n = 1$ **return** a
else return $DivConqPower(a, \lfloor n/2 \rfloor) * DivConqPower(a, \lceil n/2 \rceil)$