Mystery Code I

In line 05 the procedure MA is calling itself on a version of the input list with the kth element (a_k) removed. Assume it takes constant time to temporarily remove a_k from the list. (Doing this in constant time actually requires some extra details that we are hiding for clarity.)

00 MA($a_1, \ldots, a_n$) : list of n positive integers, $n \geq 3$)
01 if ($n = 3$) return $a_1 + a_2 + a_3$
02 else
03 bestval = 0
04 for $k = 1$ to n
05 newval = MA($a_1, a_2, \ldots, a_{k-1}, a_{k+1}, \ldots a_n$)
06 if (newval > bestval) bestval = newval
07 end for
08 return bestval

(a) Describe (in English) what MA computes.

(b) Suppose that $T(n)$ is the running time of MA on an input array of length n. Give a recursive definition of $T(n)$.

(c) How many leaf nodes are there in the recursion tree for $T(n)$? Briefly explain.

(d) Does MA run in $O(2^n)$ time? Briefly explain why or why not.

Solution:

Mystery Code II

Consider an array of n *distinct* real numbers $a_1, a_2, \ldots, a_n$. We say that the array has a *peak* at position k if the following two conditions hold for every position j between 2 and n:

(1) If $j \leq k$, then $a_{j-1} < a_j$.

(2) If $j > k$, $a_{j-1} > a_j$.

Consider the following procedure to determine position of the peak of an array (assume that the array does indeed have a peak):

```
00 procedure Find(a_1, a_2, ..., a_n: array of real numbers)
01    if (n = 1)
02       return 1
03    if (a_1 > a_2)
04       return 1
05    else if (a_n > a_{n-1})
06       return n
07    k = floor((1+n)/2)
08    if (a_{k-1} > a_k)
09       return Find(a_1, ..., a_{k-1})
10    else if (a_k < a_{k+1})
11       return Find(a_{k+1}, ..., a_n) + k
12    else
13       return k
```

(a) Consider the array $-1, 3, 6, 7, 0$. Trace the execution of the above pseudocode and show that it correctly returns the position of the peak.

(b) At line 07, what is the smallest value that n might contain? Why?

(c) Let $T(n)$ be the worst-case running time of the above pseudocode when the array has size n. Write a recurrence for $T(n)$, including the necessary base case(s). Assume that splitting the array (lines 09 and 11) takes constant time.

(d) What is the tightest big-O running time of Find?

Solution: