MATH 1P67: Mathematics for Computer Science Spring 2016

Assignment 3: Recursions and Complexity
Due: Wednesday, June 29, at 19:00 (7:00 pm)

1. Double Tower of Hanoi contains $2n$ disks of n different sizes, with two disks of each size. You must move all $2n$ disks from one peg to another, but you may move only one disk at a time, without putting a larger disk over a smaller one. How many moves does it take to transfer a double tower from one peg to another if disks of equal size are indistinguishable from one another? Find a recurrence relation for the number of moves. Then, solve the recurrence relation.

2. Below is pseudocode for a modified merge sort algorithm. This new algorithm partitions the list into four sublists instead of the usual two:
procedure newmergesort$\big(a[1, ..., n]\big)$
input:
output:
if $n > 1$ **then**
$L_1 = \text{merge}\Big(\text{newmergesort}\big(a[\ 1, ..., \lfloor n/4 \rfloor\ ]\big), \text{newmergesort}\big(a[\ \lfloor n/4 \rfloor + 1, ..., \lfloor n/2 \rfloor\ ]\big)\Big)$
$L_1 = \text{merge}\Big(\text{newmergesort}\big(a[\ \lfloor n/2 \rfloor + 1, ..., \lfloor 3n/4 \rfloor\ ]\big), \text{newmergesort}\big(a[\ \lfloor 3n/4 \rfloor + 1, ..., n\ ]\big)\Big)$
$\text{merge}\big(L_1, L_2\big)$

Complete the following two problems to determine if it is possible to improve the complexity of merge sort by partitioning the list into more than two lists of smaller sizes.

a) Analyze the worst-case runtime of the new merge sort (you may make reasonable assumptions about the length of the list).
b) Compare the complexity of the original merge sort with the complexity of the new merge sort.

3. Solve the following recurrences:

a) $T(n) = 7T(n-1) - 10T(n-2)$ for $n \geq 2$, $\quad T(0) = 2$ and $T(1) = 1$.

b) $T(n) = 6T(n-1) - 8T(n-2)$ for $n \geq 2$, $\quad T(0) = 4$ and $T(1) = 10$.

c) $T(n) = T(n-2)$ for $n \geq 2$, $\quad T(0) = 5$ and $T(1) = -1$.

d) $T(n) = -4T(n-1) + 5T(n-2)$ for $n \geq 2$, $\quad T(0) = 2$ and $T(1) = 8$.