CS 330 - Spring 2021 - Homework 6

Due: Thursday, April 1

Reading: Chapter 6, sections 1 and 2 of our textbook, pages 251-261.

Problems:

1. In the very first week of class we went over the pseudocode for the MIN-AND-MAX algorithm (in the reading from HW 1).
We also informally described how to write an efficient algorithm which finds the minimum and maximum elements of a list of numbers.

Here is a description of a **divide and conquer** algorithm to find the min and max of a list called DandC-MIN-AND-MAX(L):

Input: A list L with length n where n is an integer power of 2, that is n = 1,2,4,8,16...

1. Split n into 2 lists L1 and L2, each of length n/2. (Base case: If n/2=1 then output L1 and L2.)

2. Call DandC-MIN-AND-MAX(L1) whose output is min(L1) and max(L1) and DandC-MIN-AND-MAX(L2) whose output is min(L2) and max(L2).

3. Compare min (L1) and min(L2) and output the minimum of these as min(L)

4. Compare max (L1) and max(L2)and output the maximum of these as max(L)

Do problems (i), (ii) and (iii) below.

i. Carry out this algorithm on the 8 element list L = 12 4 9 6 15 2 7 3.
Show your work. (It is sufficient here to just show the tree representing the computation.)

State exactly how many comparisons are done on your example run.

ii. Define M(n) = the number of comparisons done by DandC-MIN-AND-MAX(L) where n = $|L|$.

Write a recurrence relation for M expressing M(n) in terms of M(k) where k is smaller than n.

iii. Now derive a closed form for your recurrence expressing M(n) as a function of n.
(Note: A closed form means to write M(n)≤ ...., where ... is an explicit function of n but does not contain M. See pages 210-211 of the textbook where this is explained, especially the paragraphs in the middle of page 211.)

2. For all the parts i. ii. and iii. below you should briefly justify your work.

i. Consider the recurrence relation T(n) = 3T(n/3)+2, with n a power of 3 (n = 3^k) and the base case T(1) = 0.

Compute the first 5 values of T(n): T(3), T(9), T(27), T(81), T(243).

Compute a closed form for T, and give the rate of growth of T (that is, does T have polynomial growth or O(n log n) growth or exponential growth or ...)?

ii. Consider the recurrence relation R(n) = $a_n = a_{n-2} + a_{n-1} + 3$ where base cases are R(1) = $a_1 = 1$ and R(2) = $a_2 = 2$.

Compute the first 6 values of R(n) past the base cases: R(3), R(4), ..., R(8). Show your work.

Compute a closed form for R, or if too hard, give the rate of growth of R (that is, does R have polynomial growth or exponential growth or ....)?

iii. Consider the recurrence relation S(n) = $a_n = 4a_{n-1} - 2$ where S(0)= $a_0 = 2$. Compute the first 6 values of S(n): S(1), S(2), S(3), ..., S(6) after S(0). Show your work.

Compute a closed form for S, and give the rate of growth of S (that is, is does S have polynomial growth or exponential growth or....)?