

# Homework Assignment No. 1

CSCI 4470/6470 Algorithms, CS@UGA, Fall 2019

**Due Thursday August 29, 2019**

*The answers must be word-processed or typed. You may substitute formulae and figures with hand-writings. Your submitted algorithms should be in the pseudo-code, not in any specific programming language. Answers deviating from these requirements will be returned without grading.*

*The answers must be the student's own work. Idea sharing and referencing to others' work (including those online) are not allowed. Plagiarism and other forms of academic dishonesty will be handled within the guidelines of the Student Handbook and reported to the University.*

**There are 7 questions and 140 points in total.**

1. **(20 points)** Consider the following summation  $sum(n) =$

$$\begin{aligned} &1+ \\ &1+2+ \\ &\dots \\ &1+2+\dots+(n-1)+ \\ &1+2+\dots+(n-1)+n \end{aligned}$$

which can be computed recursively.

Write in pseudo-code recursive function  $sum(n)$  with input parameter  $n$ . You are *not allowed* to use iterative statements in the pseudo-code. However, if you wish, your function may call other auxiliary recursive functions that you may design.

2. **(20 points)** Consider the following problem to find both the maximum and second maximum elements from a set of  $n$  integers stored in a list

$A[1..n]$ . One strategy to solve the problem is to first split the list into two sublists of roughly the equal size, find the maximum and second maximum elements from the first sublist and also find the maximum and second maximum from the second sublist, and then compare the 4 elements to identify the maximum and second maximum for the original list. Implement this idea into a recursive function in pseudocode. No iterative statements are allowed.

3. **(20 points)** (*Independent of the previous Q2.*) This question is about upper bound and lower bound of number comparisons used to find both the maximum and second maximum elements from a set of  $n$  numbers. Try the following challenges and explain how you arrive at the answers.

- (1). give an upper bound,
- (2). give a non-trivial upper bound;
- (3). give a lower bound;
- (4). give a non-trivial lower bound.

A *non-trivial bound* is one that is not obvious.

4. **(20 points)** Use the definition of big- $O$  to prove:

- (1)  $100n^2 = O(n^2 - 10n)$ ;
- (2)  $n^k = O(2^n)$ , for any fixed  $k \geq 1$ .

5. **(20 points)** Questions 3-4: page 62 (3rd ed), page 59 (2nd ed). Answer questions for parts  $c$  and  $e$  only.

6. **(20 points)** Use a recursion tree to determine a good asymptotic upper bound on time function  $T(n)$ , where  $T(n) = T(n/2) + n \log_2 n$  and  $T(1) = c$  for some constant  $c > 0$ . You may assume that  $n$  is always a power of 2.

7. **(20 points)** Use the substitution method to verify your answer for the above Q6.

**Homework submission should be in hardcopy. If for a reason an email submission is necessary, it needs an approval from this instructor. Submission needs to be received by 5:00pm on the due day. Thank you.**