

# CS3510-A: Design and Analysis of Algorithms, Spring 2021

## Homework-4

February 9, 2021

**DUE DATE: Tuesday, February 16, 11:59pm**

**Note-1:** Your homework solutions should be electronically formatted as a single PDF document that you will upload on Gradescope. If you have to include some handwritten parts, please make sure that they are very clearly written and that you include them as high resolution images.

**Note-2:** Please think twice before you copy a solution from another student or resource (book, web site, etc). It is not worth the risk and embarrassment.

**Note-3:** You need to **explain/justify** your answers. Do not expect full credit if you just state the correct answer.

**Note-4:** **You will get 2 extra points if you submit electronically typed solutions instead of hand-written.**

### Problem-1 (36 points)

The Master Theorem of Recurrence we covered in class is not applicable in all recursions. For instance, it does not apply on the following recursions.

Solve each of the following recursions. You do not need to give inductive proofs. We are only looking for the asymptotic run-time complexity of each recursion using the Big-Theta notation.

a)  $T(n) = 49T(\frac{n}{25}) + n^{(3/2)} * \log n$

b)  $T(n) = T(n - 1) + c^n$ , where  $c > 1$  is a constant

c)  $T(n) = 2T(n - 1) + 1$

### Solution

## Problem-2 (34 points)

In Lesson 8.2, Constantine described at a high-level a divide-and-conquer linear-time algorithm to compute the majority element of an array.

In this exercise, you are asked to write detailed pseudocode for that algorithm. Your algorithm should cover the case that either the original array  $A$  or the reduced array  $A'$  has an odd number of elements. It should also cover any other corner cases you can think of.

## Solution

### Problem-3 (30 points)

Design an algorithm to multiply any  $n$ -bit integer with any  $m$ -bit integer where  $n \geq m$  in  $O(n m^{\log_2(3)-1})$  time.

**Solution**