WEEK 05

1. Preparation for Assignment

If, and *only if* you can truthfully assert the truthfulness of each statement below are you ready to start the exercises.

1.1. Reading Comprehension Self-Check.

- I know why it is **false** to say that *divide-and-conquer* is a general algorithm design technique that starts solving a problem's instance by dividing it into several smaller instances, ideally of unequal size.
- I know that the Master Theorem establishes the order of growth of the solutions to the general recurrence $T(n) = aT(n/b) + f(n)$ that the running time of many divide-and-conquer algorithms satisfies.
- In addition, I know why it is **false** to say that the Master Theorem gives **explicit** solutions to this general recurrence.
- I know that *mergesort* and *quicksort* are two divide-and-conquer sorting algorithms both of whose best-case time efficiency is "linear logarithmic".
- I know that *decrease-and-conquer* might be considered a degenerate case of *divide-and-conquer*, but it is better to consider them as two different design paradigms.
- I know how to fill in the table below, that is, I can compare (by matching the correct phrase with an X in the correct table cell, only one X per each row and per each column) the orders of growth of two functions g(n) and f(n) when the ratio of g(n) to f(n), (i.e., g(n)/f(n), in the limit as n goes to infinity), approaches zero, or else some positive constant, or else infinity.

  $GR(a(n))$ is a function that returns the Growth Rate of a function.

$\lim_{x\to\infty} \frac{f(x)}{g(x)}$	$GR(g(n)) = GR(f(n))$	$GR(g(n)) > GR(f(n))$	$GR(g(n)) < GR(f(n))$
0			
$k \in \mathbb{N}, k \neq 0$			
∞			

1.2. Memory Self-Check.

1.2.1. *Applying the Master Theorem.* By filling in the table (the first row is done for you), show that you know how to use the Master Theorem, page 197 and Master Theorem, to find the Θ order of growth for solutions of the following recurrence

Date: October 18, 2018.

relations (where in every case, $T(1) = 1$):

	$T(n) =$	a	b	d	$a <, =, \text{or} > b^d$	Θ
1.	$2T(n/2) + n - 1$	2	2	1	$=$	$n \log_2 n$
2.	$2T(n/2) + 2n + 1$					
3.	$2T(n/2) + 1$					
4.	$3T(n/3) + n^2 + 2n + 1$					
5.	$4T(n/2) + n$					
6.	$4T(n/2) + n^2$					
7.	$4T(n/2) + n^3$					

2. Week 04 Exercises

2.1. **Exercise 1 on page 174.**

2.2. **Exercise 1 on page 181.**

2.3. **Exercise 5 on page 185.**

2.4. **Exercise 6 on page 186.**

2.5. **Exercise 2 on page 217.**

2.6. **Exercise 1 on page 191.**

3. Week 05 Problems

3.1. **Exercise 11 on page 175.**

3.2. **Exercise 9 on page 186.** Make sure you write out the full mathematical proof.

3.3. **Exercise 11 on page 186.**

3.4. **Exercise 12 on page 198.**