COT 5407 Introduction to Algorithms
Homework 1

Due on Tuesday, September 11, 2018, at the beginning of the class

---

1. **[5 points]** Read chapter 1 from Cormen 3rd edition and briefly (in a paragraph) answer questions **1.1-1, 1.1-3, 1.1-4**. For the reading part do not write anything in your solution.

2. **[15 points]**

   Sort the following functions in the table below from asymptotically smallest to asymptotically largest (indicate if there are ties). Do not write the proofs (but do them for practice), just the sorted list of functions. Remember than we use the convention $lg\, n = log_2\, n$.

n	$lg\, n$	$\sqrt{n}$	5^n
$\sqrt{lg\, n}$	$lg\, \sqrt{n}$	$5^{\sqrt{n}}$	$\sqrt{5^n}$
$5^{lg\, n}$	$lg\, (5^n)$	$5^{lg\, \sqrt{n}}$	$5^{\sqrt{lg\, n}}$
$\sqrt{5^{lg\, n}}$	$lg\, (5^{\sqrt{n}})$	$lg\, (\sqrt{5^n})$	$\sqrt{(lg\, (5^n))}$

3. **[15 points]** Give a solution to the following recurrences. Give tight asymptotic bounds in the form of $\Theta(f(n))$ for some recognizable function $f(n)$. If your solution needs a particular base case mention it. :

   - $T(n) = T(\sqrt{n}) + n$
   - $T(n) = T(n-1) + lg\, n$
   - $T(n) = 4T(n/8) + \sqrt{n}$
   - $T(n) = 4T(n/2) + nlg\, n$

4. **[15 points]** You will be given a stack of n pancakes that have different sizes. You need to sort the pancakes in such a way that the smaller pancakes are on top of the larger pancakes. The operation that you can perform is a **flip**. In a flip you insert a spatula under the top k pancakes for some integer k between 1 and n and flip them all over.

   (a) Describe an algorithm that sorts any stack of n pancakes with few flips as possible in the worst case.

   (b) Exactly how many flips does your algorithm do in the worst case?

5. **[15 points]** An inversion in an array $A[1..n]$ is defined as a two of indices (i, j) such that $i < j$ and $A[i] > A[j]$. The number of inversions on an array of size n is between 0 (when the array is sorted) and $\binom{n}{2}$ (when the array is sorted in reverse). Design and analyze an algorithm to count the number of inversions in an array with n elements in $O(nlog\,n)$ time. *Hint: Try modifying the Mergesort algorithm*.

6. **[10 points]** Propose a $\Theta(nlg\,n)$ time algorithm that can determine whether there are two elements in an set S whose sum is exactly a value t.

7. **[10 points]**

   Using a *Proof by Induction* prove that you can give any amount of change larger than 29 cents if you are given an unlimited supply of 6-cent coins, 10-cent coins, and 15-cent coins

8. **[15 points]** Carefully read pages 18-20 and 31-33 from Cormen 3rd edition where the correctness of insertion sort and merge sort are presented. Solve problem **2-2** (pg 40). For the reading part do not write anything in your solution.