CS 161: Recitation 1 (Fall 2016)

Question 1

1. Order the following functions so that $f_1 = O(f_2), f_2 = O(f_3), \ldots, f_6 = O(f_7)$:

$$2^n \qquad n \log n^2 \qquad n^2 \log n \qquad n^{1/4} \qquad (\log n)^4 \qquad 4^n \qquad 2^{100^{100}}$$

2. Each row in the following table describes the running time of two different algorithms that solve the same problem. For each row, indicate whether one of the algorithms will always be faster for sufficiently large n.

Algorithm 1	Algorithm 2
$O(n^2)$	$O(n^3)$
$\Omega(n^2)$	$\Omega(n^3)$
$\Theta(n \log n)$	$\Omega(n^2)$
$\Theta(n \log n)$	$O(n)$
$\Theta(n \log n)$	$\Omega(n)$

Question 2

Recall that an undirected graph is connected if it has a path connecting any two vertices, and that a tree is defined as an acyclic connected undirected graph. Prove that for a tree, $G = (V, E)$, $|V| = |E| + 1$.

Question 3

In lecture, you saw how to solve the recurrence relation for MergeSort by drawing its recursion tree and adding up the total running time. Use this method to solve the following recurrences — that is, get a tight bound of the form $T(n) = \Theta(f(n))$ for the appropriate function f.

Show your work. If you wish, you may assume that n initially has the form $n = a^i$, for an appropriate constant a.

1. $T(n) = T(n/5) + 64n$
2. $T(n) = 4T(n/4) + n$
3. $T(n) = T(n-2) + 9n$