1. **(Asymptotic Analysis)** List at least one reason why Asymptotic Analysis is a helpful tool for reasoning about algorithm runtimes in this class (as opposed to just giving runtimes in terms of exact numbers of operations).

2. **(Recursion Tree Basics)** Suppose you’re drawing a recursion tree to describe a Divide-and-Conquer algorithm. This recursive algorithm takes an input of size $n$ and recurses on 9 subproblems, each of size $\frac{n}{3}$, until it hits a base case of size 1. You may assume $n$ is a perfect power of 3.

2.1. **How many levels are in the tree?** Note: we count the root (Level 0) as a level (for example, as shown in Lecture 1, there were $1 + \log_2 n$ total levels in Karatsuba’s recursion tree).

2.2. **What is the size of a single subproblem at an arbitrary level $t$ in the tree?** Assume we’re numbering our levels using 0-indexing, as we’ve done in lecture.

2.3. **How many subproblems are at the lowest level?** (The lowest level is the level where all subproblems are now size 1).

3. **(Big-O)** Which of the following are $O(n^2)$? Please select all that apply. (As extra practice, think about how you might write a Big-O proof or disproof for each.).

   (1) $f_1(n) = 3n^2$
   (2) $f_2(n) = n \log n$
   (3) $f_3(n) = 2^{100}$
   (4) $f_4(n) = \frac{1}{1000} n^3 + n^2$

4. **(Big-Omega)** Which of the following are $\Omega(n^2)$? Please select all that apply. (As extra practice, think about how you might write a Big-Omega proof or disproof for each.).

   (1) $f_1(n) = 3n^2$
   (2) $f_2(n) = n^2 - n$
   (3) $f_3(n) = 1000 \log n$
   (4) $f_4(n) = \frac{1}{1000} n^3 + n^2$

5. **(Algorithm Runtimes)** List the Big-O runtime (in terms of $n$) of the following algorithms.

   5.1. **Karatsuba’s Integer Multiplication Algorithm** (multiplying 2 integers, each with $n$ digits)
   5.2. **Insertion Sort** (sorting an array containing $n$ elements)
   5.3. **MergeSort** (sorting an array containing $n$ elements)