(20 pts) 1. Each recurrence below solves to one of the following:
   A. $\Theta(\log n)$, B. $\Theta(n)$, C. $\Theta(n \log n)$, D. $\Theta(n^2)$, E. $\Theta(n^3)$, F. $\Theta(3^{n/5})$, G. $\Theta(5^{n/3})$, H. $\Theta(2^n)$
   
   For each one, write down the letter of the correct answer. Assume the base case $T(x) = 1$ for $x \leq 5$.

   - $T(n) = 3T(n/3) + 3n + 1$
   - $T(n) = T(n/3) + 2T(n/4) + n$
   - $T(n) = T(n - 2) + n^2 + n$
   - $T(n) = 5T(n - 3)$

(40 pts) 2. (a) Suppose we are given an unsorted list of $n$ elements and want to output the middle $n/3$ of them (everything between the $n/3$rd smallest and the $n/3$rd largest). This can be done in time: (pick the best possible using a comparison-based algorithm)

   \[ O(n) \quad O(n \log n) \quad O(n^2) \quad O(n^3) \]

   (b) Suppose we are given an unsorted list of $n$ elements and want to output the middle $n/3$ of them in sorted order. This can be done in time: (pick the best possible using a comparison-based algorithm)

   \[ O(n) \quad O(n \log n) \quad O(n^2) \quad O(n^3) \]

   (c) Draw the result of inserting the letters A,L,G,O into an initially-empty B-tree with $t=2$ (i.e., a 2-3-4 tree).
(d) What is the expected number of times the string “11” appears in a random 10-bit string? Formally, what is the expected number of positions $i$ such that the $i$th position in the string is 1 and the $(i+1)$st position in the string is 1?

(e) Give an ordering of the elements \{1, 2, 3, 4, 5, 6, 7\} such that if they were inserted in your order into a standard (simple) binary search tree, then the resulting tree would be perfectly balanced.

(40 pts) 3. In this problem we will look at two different deterministic Quicksort algorithms, each with a different rule for choosing the pivot.

(a) Suppose we choose the pivot to be the median of the $n$ numbers in the array. Assuming that it takes $O(n)$ time to compute the median (e.g., we could use the algorithm from class), what is the worst-case running time of this version of Quicksort? Give a short explanation.
(b) Suppose we choose the pivot to be the *average* of the $n$ numbers in the array (i.e., we add up all the numbers and divide by $n$). Assuming that it takes $O(n)$ time to compute the average, what is the worst-case running time of this version of Quicksort? Describe what an array would look like that achieves your $\Omega()$ bound.