CS130A Homework 1
Due 2019-07-09 by 4:59PM

1) Recurrence relationships are important. Use brute force or the master theorem to find the Big-Theta runtime of functions with the following recurrence relations:
   a) T(n) = 1 + T(n-1)
   b) T(n) = n + T(n-1)
   c) T(n) = 3*T(n/3) + 3*n
   d) T(n) = 3*T(n/2) + n
   e) T(n) = 1*T(n/2) + n^3
   f) T(n) = 2*T(n/2) + $2n^2$
2) For each of the following g(n), which are legitimate Big-O, Big-Theta, or Big-Omega for f(n) = n^2 + 2n. List all that apply.
   a) n^2
   b) n^3
   c) n
   d) 2^n
   e) lg n
3) The common earthworm has 36 chromosomes. These are paired off, much like a human. Of the 18 chromosome groups, suppose there exist 3 variants. I claim that I have a function that outputs all possible variations of the earthworm genome. Furthermore, I claim that my function runs in O(n^2) time. Could I possibly be telling the truth? Explain. Hint: What is the input?
4) A good hash function has few collisions. Why, then, do many object oriented programming languages use the memory location of an object as a default hash code for instances of a class? Note: most hash tables will be smaller than addressable memory.
5) The default hash code for strings in Java is described here: https://docs.oracle.com/javase/7/docs/api/java/lang/String.html#hashCode()
   a) List 3 properties of this that make it a good hash code.
   b) How many unique hash codes are there?
   c) Assuming a random distribution of strings (and corresponding hash codes), how many strings do you need before the probability of a collision is greater than 50%?