CS 535 Homework 7
Due: November 5 (W), in class.

Undergrads, please answer problems 1, 2, 3. Graduate students, answer all problems.

1. Consider the following sorting algorithm:

   \[ \text{Sort}(A[0 \ldots n-1]) \]

   if \( n = 2 \) and \( A[0] > A[1] \)
   
   swap \( A[0] \) and \( A[1] \)

   else if \( n > 2 \)
   
   \( m = \lceil \frac{2n}{3} \rceil \)

   Sort(A[0 \ldots m-1])

   Sort(A[n+m \ldots n-1])

   Sort(A[0 \ldots m-1])

   a. First, try out the algorithm on some examples. (No need to submit this part.)

   b. Prove that the algorithm works. That is, argue that it is a sorting algorithm. If you don’t know where to begin, think about how you would convince your future boss that the above algorithm is correct.

   c. Determine its running time. That is, draw out the recursion tree. Determine the non-recursive work performed at each level. Finally add them all up. It might be helpful to know the following property: \( a^{\log_b n} = n^{\log_b a} \).

2. C-4.14 (A simple bucket sort will obviously not work here because it will take \( O(n^2) \) time.)

3. C-4.19

4. C-4.15. (Don’t assume that \( k \) is a constant. That is, if you’re not careful, it can be a factor in the running time.)