Homework Assignment #3
Due: January 31, 2020 at 2:30 p.m.

1. In this question we look at various ways to multiply two natural numbers x and y, given to us in binary representation, if x and y have different lengths. Let m and n be the number of bits needed to represent the two numbers x and y, respectively. Suppose $m \leq n$. **All running times in this question measure the number of bit-level operations.**

In class, we saw that if $m = n$, then Karatsuba's algorithm (we called it mult3) can multiply x and y in $\Theta(n^{\log_2 3})$ time.

(a) We looked at the standard grade-school algorithm for multiplication in class. (We called it mult2.) Which of the two numbers should you halve in each iteration of the loop? Give the running time of this algorithm. State your answer in terms of m and n using Θ notation and briefly justify your answer.

(b) We can apply Karatsuba's algorithm directly to multiply x and y, even if m is strictly less than n, by modifying the inputs a little. Briefly explain how to do this and state the running time of your solution.

(c) If $m < n$, we could break y into pieces of m bits each, multiply the pieces by x and reconstruct the product xy from those pieces. Give pseudocode for this approach. Include pre- and post-conditions. If you use a loop, then state a loop invariant. You do not have to prove your algorithm is correct. Give the running time of your algorithm in terms of m and n. Explain carefully why your time bound is correct. To make the algorithm as efficient as possible, this may require you to think carefully about how the bits are represented and how you do addition.

(d) Discuss which of the algorithms of (a), (b) and (c) is best, in terms of running time, for large m and n. Under which situations would one of them be best? (For example, you might say something like "When m is $O(\sqrt{\log n})$, it is better to use the algorithm of part (?) because ...")

2. In class, we looked at a divide-and-conquer algorithm to compute x^y for natural numbers x and y (not both 0) based on the recurrence

$$x^y = \left\{ \begin{array}{ll} 1 & \text{if } y = 0 \\ x^{\frac{y}{2}} \cdot x^{\frac{y}{2}} & \text{if } y \text{ is even and greater than 0} \\ x^{\frac{y-1}{2}} \cdot x^{\frac{y-1}{2}} \cdot x & \text{if } y \text{ is odd} \end{array} \right\}.$$

Let m and n be the number of bits needed to represent the two numbers x and y, respectively. If we use Karatsuba's multiplication algorithm, how many bit-level operations are required to compute x^y in this way? Give your answer in terms of m and n, using O notation and show your answer is correct.