

In lecture, we described an algorithm of Karatsuba that multiplies two n -digit integers using $O(n^{\lg 3})$ single-digit additions, subtractions, and multiplications. In this lab we'll look at some extensions and applications of this algorithm.

1. Describe an algorithm to compute the product of an n -digit number and an m -digit number, where $m < n$, in $O(m^{\lg 3-1}n)$ time. *Hint: Break up the bigger number into chunks with m bits each.*
2. Describe an algorithm to compute the decimal representation of 2^n in $O(n^{\lg 3})$ time. (The standard algorithm that computes one digit at a time requires $\Theta(n^2)$ time.) Does the algorithm run in time that is polynomial in the input size? Does it run in time polynomial in the output size?
3. Describe a divide-and-conquer algorithm to compute the decimal representation of an arbitrary n -bit binary number in $O(n^{\lg 3})$ time. [*Hint: Let $x = a \cdot 2^{n/2} + b$. Watch out for an extra log factor in the running time.*]

Think about later:

4. Suppose we can multiply two n -digit numbers in $O(M(n))$ time. Describe an algorithm to compute the decimal representation of an arbitrary n -bit binary number in $O(M(n) \log n)$ time.