Basic Algorithms, Assignment 3

Due, Thursday, Sept 27

1. Page 67 Exercise 1 (Think of n large. Possible answers are: Negligible effect, takes (roughly) twice as long, takes much much longer.)

2. Page 67 Exercise 2

3. Page 67 Exercise 3

4. Page 68 Exercise 6

5. Let $A[i]$, $0 \leq i \leq n$, represent an $n+1$-digit decimal number α Give an algorithm (repeating what you learned in fifth grade) for dividing α by 7 (just for definiteness). Your output should be the quotient as an array $Q[i]$ and a remainder R. How long does this algorithm take in ThetaWorld. (e.g.: $\Theta(n), \Theta(n \lg n), \Theta(n^2)$ or what?)

6. Let n be a power of two, $n = 2^t$. Let p be a prime number. Our object is to calculate $7^n \bmod p$. We think of p as being of moderate size but n being very large. (For example, $p = 137$, $n = 2^{30}$.) *Assume* that we have a subroutine that multiplies two numbers modulo p in constant time.

   (a) `Method I:` Start with 1 and multiply by 7 until reaching 7^n. How long does this take. (Answer in ThetaWorld as a function of n.)

   (b) `Method II:` Start with 7 and square until reaching 7^n. How long does this take. (Answer in ThetaWorld as a function of n.)

If you take a number and double it and double it again and then double it a few more times, the number gets bigger and bigger and goes higher and higher and only arithmetic can tell you what the number is when you quit doubling.
from *Arithmetic* by Carl Sandburg