

CSE 311: Foundations of Computing I

Assignment #4

April 23, 2014

due: Wednesday, April 30, 1:30 p.m., *before lecture begins*

Bundles: The problems in each homework assignment will be divided into 2 groups (to facilitate distribution to grading TAs). You will turn in 2 corresponding bundles. Write your name in the *upper left corner* of each bundle's top page, with your last name printed clearly in CAPITAL LETTERS. Each bundle should be stapled separately. We don't supply the stapler.

This week's turnin bundles: (A) problems 1–2, (B) problems 3–4.

Textbook numbering labeled “6th edition” refers to the textbook's Sixth Edition. Numbering that is unlabeled refers to the Seventh Edition.

1. Use Euclid's algorithm to compute the following, showing the values of x and y for each iteration of the algorithm.
 - (a) $\gcd(1020, 1173)$
 - (b) $\gcd(1019, 1173)$
2. Suppose that you want to compute $\gcd(a, b)$, where a and b each have n digits. The naive algorithm that first finds the prime factorization of a and b uses approximately $10^{n/2}$ integer divisions to do so, by trying all possible divisors up to \sqrt{a} and \sqrt{b} , respectively. In contrast, Euclid's algorithm uses approximately $5n$ divisions. Suppose you were running these two algorithms on a computer that could do 10^9 integer divisions per second. Assume for simplicity that you can ignore the time for the algorithms to do anything other than integer divisions. Put your answers to the following questions into a single 3×2 table:
 - What is the greatest number n of digits that you could handle by each of the two methods in 10^{-5} seconds of computer time?
 - What is the greatest number n of digits that you could handle by each of the two methods in 1 second of computer time?
 - What is the greatest number n of digits that you could handle by each of the two methods in 10^5 seconds of computer time? (This is approximately 1 day of computer time.)
3. Section 4.3, exercise 50 [6th edition: Section 3.5, exercise 32]. Give a careful proof.
4. In this problem, you will use the RSA cryptosystem to do some encryption and decryption. Please use the version presented in lecture. Your primes are $p = 29$ and $q = 41$, and you will encrypt and decrypt two-letter messages at a time.

You will need either a calculator or computer program. I did the calculations in about 15 minutes with my calculator. Alternatively, it's fine if you decide to write a simple program to do the modular exponentiation, as long as you do it by the "repeated squaring and reduction" method used in lecture, and print out the intermediate results as described below.

- (a) What are the values of the public key n and the secret key s that correspond to the choices of p and q above?
- (b) You want to encrypt the two-letter message HI. You translate this into the integer 0809, since H is the 8th letter and I the 9th letter. Compute $C = E(809)$, showing the intermediate results after each reduction mod n .
- (c) For the value of C that you obtained in part (b), compute $D(C)$, showing the intermediate results after each reduction mod n .
- (d) There is an easy test to check if you got the right answer in part (c). It's probably a good idea to do it.
- (e) Why are these choices of p and q insufficient for encrypting and decrypting all possible 2-letter messages using the method of parts (b) and (c)? Describe a simple fix that would allow you to stick with these choices of p and q and yet be able to encrypt and decrypt all possible 2-letter messages.