Algorithms – CMSC-27200
http://alg15.cs.uchicago.edu
Homework set #6 due Wednesday, February 18, 2015

Read the homework instructions on the website. The instructions that follow here are only an incomplete summary.

Hand in your solutions to problems marked "HW." Do not hand in problems marked "DO." If you hand in homework marked **"XC" (extra credit)**, do so on **separate and separately stapled sheets, please**. PRINT YOUR NAME and SECTION NUMBER ON EVERY SHEET you submit. **Use LaTeX to typeset your solutions**. Hand in your solutions on paper, do not email.

When writing pseudocode, **explain the meaning of your variables.** Use comments to explain what is happening in each line. Also, give a short explanation of the idea behind the algorithm. **Unless otherwise stated, describe your algorithms in pseudocode. Elegance of your code matters.**

Carefully study the policy (stated on the website) on collaboration, internet use, and academic integrity. **State collaborations and sources both in your paper and in email to the instructors.**

6.1 DO **(Euclid's algorithm and multiplicative inverse)** Study the new handout by this title. View all exercises stated in the handout as "DO" exercises; solve them. Some of them will appear as HW exercises below. We refer to the handout as the "Euclidean" handout.

6.2 DO **(Least common multiple)** Study the definition of least common multiple: We say that m is a least common multiple of a and b if

(i) $a \mid m$ and $b \mid m$ (: m is a common multiple :)

(ii) if $a \mid e$ and $b \mid e$ then $m \mid e$ (: m is a divisor of all common multiples :)

6.3 DO **(Least common multiple exists)**: Prove that for every a and b, such an m exists. Do not use unique prime factorization.

Hint. If $a = 0$ or $b = 0$ then $m = 0$ satisfies the conditions.

If $a, b \neq 0$ then let $d = \gcd(a, b)$; note that $d \neq 0$. Let $m = ab/d$. Prove that (1) m is an integer and (2) m satisfies conditions (i) and (ii). To prove (ii), first assume $e \mid ab$. In this case, the condition $ab/d \mid e$ is equivalent to saying that $ab/e \mid d$. What are the divisors of d ?

If e is not a divisor of ab, let $f = \gcd(e, ab)$. Then f is a common multiple of a and b, and also divides ab, so we can use the result we just proved.

6.4 DO: To make the lcm notation unique, we also require $\operatorname{lcm}(a, b)$ to be non-negative. (a) With this convention, prove that

$$(\forall a, b)(\gcd(a, b)\operatorname{lcm}(a, b) = |ab|). \tag{1}$$

(b) If a, b are relatively prime (i. e., $\gcd(a, b) = 1$) then $\operatorname{lcm}(a, b) = |ab|$.

6.5 DO: (a) Let a, b be positive integers, $a = \prod_i p_i^{k_i}$ and $b = \prod_i p_i^{\ell_i}$ where the p_i are prime numbers and $k_i, \ell_i \geq 0$. Let $\gcd(a, b) = \prod_i p_i^{r_i}$ and $\operatorname{lcm}(a, b) = \prod_i p_i^{s_i}$. Express r_i and s_i in terms of k_i and ℓ_i.
(b) **XC (3 points)** Prove that the gcd and lcm operations are mutually distributive, so for instance $\operatorname{lcm}(\gcd(a, b), c) = \gcd(\operatorname{lcm}(a, c), \operatorname{lcm}(b, c))$.

6.6 DO: Recall the definition of congruence: We say that $a \equiv b \pmod{m}$ if $m \mid a - b$. Prove: If $a \equiv x \pmod{m}$ and $b \equiv y \pmod{m}$ then

(a) $a + b \equiv x + y \pmod{m}$

(b) $a - b \equiv x - y \pmod{m}$

(c) **HW (3 points)** $ab \equiv xy \pmod{m}$. (You may use parts (a), (b) and the fact that congruence mod m is an equivalence relation.) Hint: prove that $ab \equiv ay \pmod{m}$ and $ay \equiv xy \pmod{m}$.

6.7 DO: Prove that the following two statements are equivalent:

(A) $a \equiv b \pmod{r}$ and $a \equiv b \pmod{s}$

(B) $a \equiv b \pmod{m}$ where $m = \operatorname{lcm}(r, s)$.

6.8 **HW (4+2 points) (Euclid's rounds)** Solve Exercise 2.3 of the "Euclidean" handout.

6.9 **HW (3+3 points)** (a) Compute $21^{-1} \bmod 76$. Use the method described in the "Euclidean" handout, Section 7. Do not use electronic devices. Show each step of your work. (b) Compute $\gcd(228, 63)$. Write this number as a linear combination of 228 and 63. (Read Exercise 7.2 of the "Euclidean" handout for a hint.)

6.10 **HW (12 points) (Multiplicative inverse: pseudocode)** Solve Exercise 7.1 of the "Euclidean" handout. Use the method illustrated on an exmaple in Section 7 of the handout. Substantially different methods will not be accepted.

6.11 DO **(Fermat's little Theorem)**: Study Fermat's little Theorem: If p is a prime and $\gcd(a, p) = 1$ then $a^{p-1} \equiv 1 \pmod{p}$.

Prove it for $p \leq 7$.

6.12 **(HW, 3 points) (application of Fermat's little Theorem)**: Compute ($7^{10^9} \bmod 101$) (the smallest non-negative residue of 7^{10^9} modulo 101, see the Euclidean handout, paragraph after Exercise 1.13). Do not use any electronic devices; in fact, no calculations are needed at all. **Show each step of your work.** (Note: the exponent is 10^9, or one billion.)

6.13 DO: Study the RSA crypotosystem from the text.

6.14 **HW (5 points)** Let p, q be the primes Alice uses for her RSA keys. So she makes the number $N = pq$ public. She also computes the number $K = (p-1)(q-1)$, uses it to produce the rest of her keys, and throws K in the garbage. Gabriel, a garbage analyst, finds K in Alice's trash bin. While this suffices for Gabriel to break Alice's keys, for private reasons he also wants to know p and q. Prove: Given N and K, Gabriel can compute p and q in polynomial time. Describe Gabriel's algorithm in unambiguous English, no pseudocode required. Prove that the algorithm runs in polynomial time.

6.15 **XC (5 points, due Feb 25)** Alice generates her RSA key using the primes p and q. Given $N = pq$ and any multiple of $K = (p-1)(q-1)$, show how to decrypt the messages sent to Alice in polynomial time.

6.16 **HW (3 points)** (This problem was updated 2-17 3:45am: in line 1 of the pseudocode, $<$ was replaced by $\leq$.)

Frank wants to find out whether or not a given number $b \geq 10$ is prime. He uses the following algorithm:

```
0    x := 2        [initialize]
1    while  x^2 ≤ b do
2        if x | b then break, print "b composite"
3        else  x := x + 1
4    end(while)
5    print "b prime"
```

Is this a polynomial-time algorithm? Reason your answer.

6.17 **HW (3 points)** Ashwin and Ming use the same software, OS, and hardware to produce "random" primes for their RSA keys. Ashwin receives primes (p, q) and Ming receives primes (p, r). Prove that if they use these pairs of primes to construct their RSA keys, neither of them will be secure – unauthorized third parties can easily decrypt the messages sent to either of them using these keys. (This actually seems to be happening with non-negligible frequency in the world. In fact, there is a set of nine large primes such that the product of each of the $\binom{9}{2} = 36$ pairs have been used as public keys. Of course all the 36 keys are compromised.)

6.18 DO: Study (from the text) how to use a public key cryptosystem to produce digital signatures.