

1 Schedule

The homework is **due Sep 14**.
The **QUIZ** will be on **Thursday, Sep. 16**.

2 List of algorithms covered in the class

(B-basic, I-intermediate, A-advanced):

- B: Addition (p.11, DSV).
- B: Multiplication (p.15, DSV).
- B: Division (p.15, DSV).
- B: Modular exponentiation (p.19, DSV).
- B: Euclid's algorithm (p.20, DSV).
- I: Extended Euclid's algorithm (p.21, DSV).
- A: Primality testing (p.25, DSV).
- A: Generating random primes (p.28, DSV).
- A: RSA (p.33, DSV).

3 Basic material

Important concepts, problems, theorems, and algorithms:

- Modular arithmetic, Fermat's little theorem.

Theorem: Let p be a prime and let a be an integer such that $\gcd(a, p) = 1$. Then $a^{p-1} \equiv 1 \pmod{p}$.

Theorem: Let p be a prime and let a be an integer. Then $a^p \equiv a \pmod{p}$.

Testing method:

- Compute $a^b \pmod{c}$. (c will be a prime smaller than 20.)
- Trace the execution of Euclid's gcd algorithm.
- Compute the multiplicative inverse of a modulo b .
- Apply Fermat's little theorem in a computation (see problems 1.1, 1.4, 1.5, below).

Example problems (solve, but do NOT turn in):

- 1.1 Compute $3^{80} \pmod{5}$.
- 1.2 Compute $\gcd(30, 81)$. Compute $\gcd(55, 34)$. Use Euclid's gcd algorithm. Show all steps.
- 1.3 Compute the multiplicative inverse of 26 modulo 677.
- 1.4 Is $4^{200} - 9^{100}$ divisible by 35? Use Fermat's little theorem to prove your answer.
- 1.5 What is $3^{3^{100}} \pmod{5}$? (as usual, a^{b^c} is a raised to the b^c -th power).
- 1.6 Prove that for every integer x , either $x^2 \equiv 0 \pmod{4}$ or $x^2 \equiv 1 \pmod{4}$.

1.7 Let p, q be two different primes. Let x, y be such that $x \equiv y \pmod{p}$ and $x \equiv y \pmod{q}$. Prove that $x \equiv y \pmod{pq}$.

1.8 For each of the following—prove or disprove (clearly state which of the two are you doing):

- For all $x \in \mathbb{Z}$ such that $\gcd(x, 19) = 1$ we have $x^{18} \equiv 1 \pmod{19}$.
- For all $x \in \mathbb{Z}$ such that $\gcd(x, 21) = 1$ we have $x^{18} \equiv 1 \pmod{21}$.
- For all $x \in \mathbb{Z}$ we have $x^{37} \equiv x \pmod{37}$.
- For all $x \in \mathbb{Z}$ we have $x^{37} \equiv x \pmod{35}$.

4 Homework

1.9 (due Sep 14) Solve the following system of congruences:

$$\begin{aligned}x &\equiv 20 \pmod{21} \\x &\equiv 21 \pmod{22} \\x &\equiv 22 \pmod{23}\end{aligned}$$

(HINT: Chinese remainder theorem.)

1.10 (due Sep 14) Let x, y be unknown positive integers. Let $A = xy$ and $B = x + y$. Give a polynomial-time algorithm which on input A, B computes x, y .

1.11 (due Sep 14) Let p be a prime and let a, b be two integers such that $a^2 \equiv b^2 \pmod{p}$. Prove that either $a \equiv b \pmod{p}$ or $a \equiv -b \pmod{p}$. (HINT: you will need to use the following fact about primes and divisibility. If p is a prime and $p \mid cd$ then $p \mid c$ or $p \mid d$.)

1.12 (due Sep 14) Let p be a prime such that $p \equiv 3 \pmod{4}$. We would like to have an algorithm which on input x computes the square root of x , that is, y such that $y^2 \equiv x \pmod{p}$. Show that we can let $y := x^{(p+1)/4} \pmod{p}$.

1.13 (due Sep 14) [BONUS PROBLEM] Professor A designed a black-box which on input a computes a^2 in time $O(\log a)$. We would like to use the black-box to multiply numbers, i. e., on input a, b we want to compute ab . We want our algorithm to run in time $O(\log(ab))$.

- Give such an algorithm.
- Suppose now, that instead of $x \mapsto x^2$ black-box, we have $x \mapsto x^3$ black-box. Show how we can use the new black-box to multiply numbers a, b in time $O(\log(ab))$.
- Suppose now, that instead of $x \mapsto x^2$ black-box, we have $x \mapsto x^4$ black-box. Show how we can use the new black-box to multiply numbers a, b in time $O(\log(ab))$.

In parts a), b), c) you can assume that we can add two numbers c, d in $O(\log(cd))$ -time. You can also assume that for any constant f we can divide d by f in $O(\log d)$ -time.

1.14 (due Sep 14) [BONUS PROBLEM] Fibonacci numbers are defined as follows: $F_0 = 0, F_1 = 1$, and $F_n = F_{n-1} + F_{n-2}$ for $n \geq 2$. Give a polynomial-time algorithm which on input n and M outputs $(F_n \pmod{M})$. (Note that the input length is $\Theta(\log n + \log M)$, and your algorithm has to run in time polynomial in the input length).

5 Additional problems from the book (do NOT turn in)

Try to solve the following problems. A few of them will be on the quiz.

- 1.1, 1.4, 1.5, 1.10, 1.11, 1.14, 1.15, 1.19, 1.20, 1.22, 1.23, 1.25, 1.26, 1.31, 1.32, 1.37, 1.39.