Crypto-refresh. Exercises sheet.

Cost analysis, Computational algebra and reductions

Exercice 1. Asymptotics

a. For each of the following functions, tell whether they are $O(n), O(n^2), O(n \log n)$, $\Omega(n)$, $\Omega(n^2)$, $\Omega(n \log n)$, $\Theta(n)$, $\Theta(n^2)$, $\Theta(n \log n)$:

1. $f(n) = 2n^2 + 3n + 5$
2. $g(n) = \frac{2n^2+5n-2}{n+1}$
3. $h(n) = 2n \log^2 n + n$
4. $i(n) = n^{1.5} \log n$

Exercice 2. Euclid's algorithm

a. Apply the extended Euclidean algorithm to compute the inverse of 21 modulo 40.

Now consider the polynomials $P = X^4 + X^3 + 2X^2 + 2$ and $Q = X^3 + X + 1$ with coefficients in the field $\mathbb{Z}/3\mathbb{Z}$.

b. Compute their GCD and the related bezout coefficients.

c. Could we have found it more quickly ?

Exercice 3. Chinese Remainder Theorem

A gang of 17 pirates has collected a treasure composed of golden coins of equal value. They decided to share it equally and give the remainder to the cook. He would then receive 3 coins. However they get into a fight where 6 pirates die. The cook would then get 4 coins. After the sinking of their boat, only six pirates and the cook are saved and the sharing would lead 5 coins to the cook.

a. What is the smallest value that the cook would get once he will manage to poison the rest of the pirates ?

Exercice 4. Fast exponentiation algorithm

We will study the cost of computing a^k where a is an element of a ring R and k a positive integer.

a. Propose a naive algorithm, named `NaiveExp`(a, k).

b. By relating a^k with $a^{\frac{k}{2}}$, propose a recursive algorithm, named `RecExp`(a, k)

c. Propose an iterative algorithm `IterExp`(a, k) (equivalent to `RecExp`(a, k) in cost), based on the binary representation of k

d. When $R = \mathbb{Z}/n\mathbb{Z}$ with $n < 2^{32}$, all basic arithmetic operations in R take $\Theta(1)$. What is the cost of the three above algorithms ?

e. Same question when n may be any integer (possibly larger than 2^{32}). Which of these algorithms can actually be run in practice for n of bit-size 128 ?

f. Same question when $R = \mathbb{Z}$.

Exercice 5. Computing a Fibonacci number

The sequence of Fibonacci is defined by the recurring formula

$$\begin{cases} u_{n+1} &= u_n + u_{n-1} \\ u_0 &= 1 \\ u_1 &= 1 \end{cases}$$

We will study the cost of computing the n-th term of the sequence. For the moment we will use the arithmetic cost model : each arithmetic operation has a unit cost.

a. Propose a recursive algorithm computing u_n. Estimate an upper and lower bound on its cost.

b. Propose an iterative algorithm and state its computational cost.

c. Let $v_n = \begin{bmatrix} u_{n+1} \\ u_n \end{bmatrix}$. Propose a relation between v_{n+1} and v_n.

d. Deduce an algorithm computing v_n and hence also u_n. What is its cost ?

e. We now consider the cost model of binary operations. What is the cost of the last algorithm ?

Exercice 6. Computing the modular inverse

Recall that the Euler theorem states that if $GCD(a, n) = 1$ then $a^{\phi(n)} = 1 \mod n$.

a. Apply it to compute $5^{2021} \mod 24$

b. Deduce from Euler theorem an algorithm computing the inverse modulo n

c. Apply it to compute $7^{-1} \mod 45$

d. How does this inversion algorithm compares to the one based on the extended Euclidean algorithm

1. assuming multiplication is quadratic ($O(n^2)$)
2. assuming multiplication is quasi-linear ($O\tilde{}(n)$)

Exercice 7. Chinese Remainder Theorem

a. compute the inverse of 49 modulo 55

Exercice 8. The Multiplicative group of $\mathbb{Z}/10\mathbb{Z}$

a. Compute $\Phi(10)$. Enumerate the elements of $(\mathbb{Z}/10\mathbb{Z})^*$ and compute the order of each of them.

b. Let g be the smallest primitive root of $(\mathbb{Z}/10\mathbb{Z})^*$, compute the table of indices with respect to g of the elements of $(\mathbb{Z}/10\mathbb{Z})^*$. To which additive group is $(\mathbb{Z}/10\mathbb{Z})^*$ isomorphic ?

c. Based on the exponentiation modulo n, give an algorithm computing x^{-1} in $(\mathbb{Z}/n\mathbb{Z})^*$, knowing $\Phi(n)$. State its cost.

Exercice 9. The field with 4 elements

a. Write down the addition and multiplication tables of $\mathbb{Z}/4\mathbb{Z}$

b. Write down the addition and multiplication tables of $\mathbb{F}_4$.

c. Is there a isomorphism between both sets ?

Exercice 10. The AES field $\mathbb{F}_{256}$

a. Explain how to define the finite field $\mathbb{F}_{256}$ with 256 elements.

b. Among the following polynomials, only one can be used for this construction, which one is it ?

1. $X - 256$
2. $X^2 + X + 1$
3. $X^{256} + X + 1$
4. $X^4 + X + 1$

5. $X^8 + X^4 + X^3 + X^2 + 1$
6. $X^8 + X^4 + X + 1$

c. Explain how you would prove that the selected polynomial is irreducible (but don't do it).

d. You have to provide a `C` implementation of $\mathbb{F}_{256}$. How would you represent an element ?

e. Write the addition and multiplication functions

Exercice 11. $\mathbb{F}_8$

a. Prove that $Q = x^3 + x + 1$ is a primitive polynomial over $\mathbb{F}_2[x]$.

b. Write down the addition, multiplication and inversion tables of the field with 8 elements.

c. Provide a `C` implementation of $\mathbb{F}_8$ using a Zech log representation

Exercice 12. Reductions

Logic Refresh :
— An implication $A \Rightarrow B$ is equivalent to its contraposition : $\neg B \Rightarrow \neg A$.
For instance *"All fire make smoke"* is equivalent to *"If there is no smoke, there is no fire."*
— A syllogism is the following reasoning :

$$\text{If } \begin{cases} A \Rightarrow B \\ A \text{ is true} \end{cases} \text{, then } B \text{ is true.} \tag{1}$$

Combining both, we get the following reasoning :

$$\text{If } \begin{cases} \neg B \Rightarrow \neg A \\ A \text{ is true} \end{cases} \text{, then } B \text{ is true.} \tag{2}$$

which in some cases is more suitable for the proof.

In cryptography, we apply this reasoning to prove that a protocol (call it Protocol 1) is secure, knowing that another protocol (call it Protocol 2) is already proven to be secure.

a. State explicitly the clauses A and B in order to make this proof.

Cryptographic hash functions are functions of the form $h : \{0,1\}^* \longrightarrow \{0,1\}^n$ transforming any string of bits of any length into a string of n bits and satisfying several properties of uniformity (which will be formalized later later on). The following security properties are defined :

(i) Resistance to the pre-image : knowing $y = h(x)$, hard to find x

(ii) Resistance to the second pre-image : knowing x, hard to find x' such that $h(x) = h(x')$.

(iii) Resistance to collisions : hard to find x and x' such that $h(x) = h(x)$.

b. Prove that (iii) implies (ii).

c. Prove that (ii) implies (i).

d. The security of the RSA asymetric cryptosystem is strongly connected to the difficulty of factoring large integers of the form $n = pq$ where p and q are prime of similar bitsize. How would one prove that "finding RSA's private key is as hard as factoring such an integer" (you don't have to do the proof).