Année universitaire 2024-2025, session 1

UE 4TMA901

Algorithmique arithmétique

Enseignants responsables : Xavier Caruso et Jean-Marc Couveignes.

Examen du jeudi 19/12/2024 à 9h00 (durée trois heures)

⋆⋆⋆

Calculette autorisée. Documents non-autorisés.

Calculators are allowed. Documents are not.

⋆⋆⋆

Ce sujet comporte quatre pages et deux parties à rédiger sur deux copies différentes.

This exam consists of two parts and four pages. Please write on two distinct papers.

Part I. Consists of 3 exercises. *Comporte 3 exercices.*

Exercise 1 :

Let $n = 7867$. We have $n - 1 = 2.3^2.19.23$. We compute.
Soit $n = 7867$. *On a* $n - 1 = 2.3^2.19.23$. *On calcule.*

$$\begin{aligned} 2^{\frac{n-1}{2}} &= -1 \bmod n \\ 2^{\frac{n-1}{9}} &= 1 \bmod n \\ 2^{\frac{n-1}{19}} &= 5437 \bmod n \\ 2^{\frac{n-1}{23}} &= 7369 \bmod n \\ 3^{\frac{n-1}{3}} &= 1465 \bmod n \end{aligned}$$

1. Which is the order of 2 mod n in the group $(\mathbb{Z}/n\mathbb{Z})^*$?
Quel est l'ordre de 2 mod n *dans le groupe* $(\mathbb{Z}/n\mathbb{Z})^*$ *?*

2. What can you say about the order of 3 mod n in the group $(\mathbb{Z}/n\mathbb{Z})^*$?
Que pouvez vous dire de l'ordre de 3 mod n *dans le groupe* $(\mathbb{Z}/n\mathbb{Z})^*$ *?*

3. Deduce that n is a prime integer.
En déduire que n *est premier.*

Exercise 2 :

We want to factor the integer $n = 45649$ using Dixon's linear sieve.
On veut factoriser l'entier $n = 45649$ *avec l'algorithme de Dixon.*

1. Recall the principle of this algorithm.
Rappelez le principe de cet algorithme.

2. We have found the following three congruences.
On a trouvé les congruences suivantes.

$$\begin{aligned} 1282^2 &\equiv 2^5.5 \bmod n \\ 5673^2 &\equiv 2^7.3 \bmod n \\ 247^2 &\equiv 2^{10}.3.5 \bmod n \\ 4241^2 &\equiv 3.5^3 \bmod n \end{aligned}$$

Terminate the algorithm. You will write the matrix of exponents, compute its kernel modulo 2, deduce a congruence between two squares modulo n then a factorisation of n as a product of two non-trivial factors. Prove that these factors are prime integers.

Terminez l'algorithme. Vous écrirez la matrice des exposants. Vous calculerez son noyau modulo 2. *Vous en déduirez une congruence entre deux carrés modulo* n, *puis une factorisation de* n *comme produit de deux facteurs non-triviaux. Et vous vérifierez que ces deux facteurs sont premiers.*

Exercise 3 :

Let $n = 2459$. One checks that $n - 1 = 2.1229$ with 1229 a prime. We set $g = 2 \bmod n$ and check that $g^{1229} = -1 \bmod n$. *Soit* $n = 2459$. *On vérifie que* $n - 1 = 2.1229$ *avec* 1229 *premier. Soit* $g = 2 \bmod n$. *On vérifie que* $g^{1229} = -1 \bmod n$.

1. Prove that g generates the group $(\mathbb{Z}/n\mathbb{Z})^*$. *Montrez que* g *engendre le groupe* $(\mathbb{Z}/n\mathbb{Z})^*$.

2. Let $h = 683 \bmod n$. We want to compute the discrete logarithm of h in base g. We denote it $\log_g h$. We pick random integers k in $[1, n-2]$ and compute $g^k h = a_k \bmod n$ with $1 \leqslant a_k \leqslant n-1$. We keep those values of k for which a_k is 3-smooth.

Soit $h = 683 \bmod n$. *On veut calculer le logarithme discret de* h *dans la base* g. *On le note* $\log_g h$. *On tire au hasard des entiers* k *dans* $[1, n-2]$ *et on calcule* $g^k h = a_k \bmod n$ *avec* $1 \leqslant a_k \leqslant n-1$. *On retient les valeurs de* k *telles que* a_k *soit* 3*-lisse.*

We find. *On trouve.*

$$\begin{aligned} g^{1466} h &= 2^2.3^2 \bmod n \\ g^{914} h &= 2^6.3^3 \bmod n \end{aligned}$$

Deduce two linear equations in the unknows $x = \log_g h$ and $y = \log_g 3$. Solve the corresponding system and compute the value of x. Note that $g = 2$ in this case.

En déduire deux relations linéaires entre $x = \log_g h$ *et* $y = \log_g 3$. *Résoudre ce système linéaire et calculer* x. *Notez que* $g = 2$.

Part II. Consists of 3 exercises. *Comporte 3 exercices.*

Exercise 4 :

We consider the following SageMath code.
On considère le code SageMath ci-dessous.

```
QC = QuantumComputer()
a = QC.malloc(1)
b = QC.malloc(1)
c = QC.malloc(1)
QC.hadamard(a)
QC.hadamard(b)
QC.CCX(a, b, c)    # a and b are the controlling bits
if QC.measure(c) == 1:
    raise RuntimeError
QC.CX(a, c)        # a is the controlling bit
QC.CX(b, c)        # b is the controlling bit
QC.X(c)
```

1. What is the probability that the above code raises a `RuntimeError` ?
Quelle est la probabilité que ce code produise un `RuntimeError` ?

2. When no error occurs, what is the internal state of the quantum computer `QC` after the execution of the above code ?
Lorsque aucune erreur n'apparaît, quel est l'état interne de l'ordinateur quantique `QC` après l'exécution du code ?

3. Is it possible to obtain the same internal state by only applying X-gates, CX-gates, CCX-gates and Hadamard gates (but no measures) ?
Est-il possible d'obtenir le même état interne en appliquant uniquement des portes X, CX, CCX et Hadamard (mais pas de mesures) ?

Exercise 5 :

We now run Shor's circuit with a biperiodic function $f : \mathbb{Z}^2 \to X$ and get the following outcomes.
On exécute le circuit de Shor avec une fonction bipériodique $f : \mathbb{Z}^2 \to X$. Les mesures réalisées donnent les résultats suivants.

1st trial / *1er essai :*	(01100110, 00110100)
2nd trial / *2ème essai :*	(00110011, 10011010)
3rd trial / *3ème essai :*	(00000000, 00000000)
4th trial / *4ème essai :*	(00000000, 00000000)
5th trial / *5ème essai :*	(11001101, 01100110)

6th trial / *6ème essai :* (00110011, 10011001)
7th trial / *7ème essai :* (11001101, 01100111)
8th trial / *8ème essai :* (10011000, 11001101)
9th trial / *9ème essai :* (11001101, 01100100)
10th trial / *10ème essai :* (11001100, 01100111)

1. Draw the above points in the plane and figure out what is the dual of the lattice of periods of f.

Dessiner les points ci-dessus dans le plan et deviner quel est le dual du réseaux des périodes de f.

2. Compute the lattice of periods of f. *Calculer le réseau des périodes de* f.

Exercise 6 :

We denote by $Q_1 = \mathbb{C}\,|0\rangle \oplus \mathbb{C}\,|1\rangle$ the $\mathbb{C}$-vector space of (unnormalized) 1-qubits. We denote by $Q_2 = \mathbb{C}\,|00\rangle \oplus \mathbb{C}\,|01\rangle \oplus \mathbb{C}\,|10\rangle \oplus \mathbb{C}\,|11\rangle$ the $\mathbb{C}$-vector space of (unnormalized) 2-qubits. Prove that there is *no* unitary transformation $f : Q_1 \to Q_2$ such that $f(q) = q \otimes q$ for all $q \in Q_1$.

On note $Q_1 = \mathbb{C}\,|0\rangle \oplus \mathbb{C}\,|1\rangle$ *l'espace vectoriel des* 1*-qubits non normalisés. On note* $Q_2 = \mathbb{C}\,|00\rangle \oplus \mathbb{C}\,|01\rangle \oplus \mathbb{C}\,|10\rangle \oplus \mathbb{C}\,|11\rangle$ *l'espace vectoriel des* 2*-qubits non normalisés. Montrer qu'il n'existe pas de transformation unitaire* $f : Q_1 \to Q_2$ *telle que* $f(q) = q \otimes q$ *pour tout* $q \in Q_1$.