Quantum Computing Exercises # 5

Ronald de Wolf

Mar 1, 2011

(to be handed in before or at the start of the lecture on Mar 8)

1. This exercise is about efficient classical implementation of modular exponentiation.

   (a) Given n-bit numbers x and N (where $n = \lceil \log_2 N \rceil$), compute the whole sequence x^0 mod N, x^1 mod N, x^2 mod N, x^4 mod N, x^8 mod N,x^{16} mod $N, \ldots, x^{2^{n-1}}$ mod N, using $O(n^2 \log(n) \log\log(n))$ steps. *Hint: The Schönhage-Strassen algorithm computes the product of two n-bit integers mod N, in $O(n \log(n) \log\log(n))$ steps.*

   (b) Suppose n-bit number a can be written as $a = a_{n-1} \ldots a_1 a_0$ in binary. Express x^a mod N as a product of the numbers computed in part (a).

   (c) Show that you can compute $f(a) = x^a$ mod N in $O(n^2 \log(n) \log\log(n))$ steps.

2. Use Shor's algorithm to find the period of the function $f(a) = 7^a$ mod 10, using a Fourier transform over $q = 128$ elements. Write down all intermediate superpositions of the algorithm. You may assume you're lucky, meaning the first run of the algorithm already gives a $b = cq/r$ where c is coprime with r.

3. Suppose we can apply a unitary U and we are given an eigenvector $|\psi\rangle$ of U ($U|\psi\rangle = \lambda|\psi\rangle$), and we would like to approximate the corresponding eigenvalue λ. Since U is unitary, λ must have magnitude 1, so we can write it as $\lambda = e^{2\pi i\phi}$ for some real number $\phi \in [0, 1)$; the only thing that matters is this phase ϕ. Suppose for simplicity that we know that $\phi = 0.\phi_1 \ldots \phi_n$ can be written with n bits of precision.

   (a) Let V be a two-register unitary that maps $|k\rangle|\psi\rangle \mapsto |k\rangle U^k|\psi\rangle$, for any n-bit integer k. Write the state that results from applying V to $\frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k\rangle|\psi\rangle$
   (to avoid trivialities, your answer shouldn't contain the letter 'U' anymore).

   (b) Give a quantum algorithm that computes ϕ exactly, using one application of V and a small (polynomial in n) number of other gates. *Hint: use the inverse QFT.*

4. This exercise explains RSA encryption. Suppose Alice wants to allow other people to send encrypted messages to her, such that she is the only one who can decrypt them. She believes that factoring an n-bit number can't be done efficiently (efficient = in time polynomial in n). So in particular, she doesn't believe in quantum computing.

   Alice chooses two large random prime numbers, p and q, and computes their product $N = p \cdot q$. She computes the so-called Euler ϕ-function: $\phi(N) = (p-1)(q-1)$; she also chooses an

encryption exponent e, which doesn't share any nontrivial factor with $\phi(N)$ (i.e., e and $\phi(N)$ are coprime). Group theory guarantees there is an efficiently computable *decryption exponent* d such that $de = 1 \bmod \phi(N)$. The *public key* consists of e and N (Alice puts this on her homepage), while the *secret key* consists of d and N. Any number $m \in \{1, \ldots, N-1\}$ that is coprime to N, can be used as a message. There are $\phi(N)$ such m, and these numbers form a group under the operation of multiplication mod N. The number of bits $n = \lceil \log_2 N \rceil$ of N is the maximal length (in bits) of a message m and also the length (in bits) of the encryption.[1] The encryption function is defined as $C(m) = m^e \bmod N$, and the decryption function is $D(c) = c^d \bmod N$.

(a) Give a randomized algorithm by which Alice can efficiently generate the secret and public key. *Hint: the prime number theorem implies that roughly* $1/\log N$ *of the numbers between 1 and* N *are prime; also there is an efficient algorithm to test if a given number is prime. Be explicit in how many bits your primes* p *and* q *will have.*

(b) Show that Bob can efficiently compute the encoding $C(m)$ of the message m that he wants to send to Alice, knowing the public key but not the private key. *Hint: exercise 1*

(c) Show that $D(C(m)) = m$ for all possible message.
Hint: the set of all possible messages forms a group of size $\phi(N)$*. Euler's Theorem says that in any group* G*, we have* $a^{|G|} = 1$ *for all* $a \in G$ *(here '*1*' is the identity element in the group).*

(d) Show that Alice can efficiently and correctly decrypt the encryption $C(m)$ that she receives from Bob.

(e) Show that if Charly could factor N, then he could efficiently decrypt Bob's message.

[1] A typical size is $n = 1024$ bits, which corresponds to both p and q being numbers of roughly 512 bits.