CS 880: Quantum Information Processing 10/4/10

Lecture 13: Factoring Integers

Instructor: Dieter van Melkebeek Scribe: Mark Wellons

In this lecture, we review order finding and use this to develop a method for factoring integers efficiently. With the exception of order finding, none of today's derivations rely on quantum computing.

1 Order Finding

In the previous class we covered order finding, which solves the following problem: Given integers $a, M > 0$ and $0 < a < M$ with $\gcd(a, M) = 1$, find the smallest integer $r > 0$ such that $a^r \equiv 1 \mod M$.

Recall from the previous lecture that we used eigenvalue estimation to developed a quantum procedure that runs in time poly-log$(M + N)$ and returns $\tilde{\omega}_j$ such that j is uniformly distributed in $\{1, 2, ...r\}$ and

$$\Pr\left[\left|\tilde{\omega}_j - \frac{j}{r}\right| \leq \frac{1}{N}\right] \geq \frac{8}{\pi^2} \approx 0.81. \tag{1}$$

Here N is defined as

$$N = 2^n, \tag{2}$$

where n is the number of qubits used in the eigenvalue estimation and with larger n comes greater accuracy. There are two important facts about this procedure that we can exploit.

1.1 Fact 1

If the eigenvalue estimation is precise enough that

$$\left|\tilde{\omega}_j - \frac{j}{r}\right| \leq \frac{1}{2M^2} \tag{3}$$

then we can recover j/r in reduced terms in time poly-log(M). By reduced terms, we mean that we can find j' and r' such that $\gcd(j', r') = 1$ and $j'/r' = j/r$.

To recover j/r in reduced terms, we use continued fraction expansion (CFE). By definition, a continued fraction takes the form

$$a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \cfrac{b_3}{\cdots}}} \tag{4}$$

where $a_i, b_i \in \mathbb{Z}$. A continued fraction is sometimes denoted as

$$\sum_{i=0}^{\infty} \frac{b_i|}{|a_i} \tag{5}$$

and the kth convergent is

$$\sum_{i=0}^{k} \frac{b_i|}{|a_i} = \frac{p_k}{q_k} \tag{6}$$

where $p_k, q_k \in \mathbb{Z}$ and $\gcd(p_k, q_k) = 1$.

To construct the CFE of some $x \in \mathbb{R}$, we write x as

$$x = \lfloor x \rfloor + (x - \lfloor x \rfloor) = \lfloor x \rfloor + \frac{1}{1/(x - \lfloor x \rfloor)}. \tag{7}$$

Since $1/(x - \lfloor x \rfloor) \geq 1$, it itself can be expanded into a CFE. Eventually, the expansion will end if for some iteration $x - \lfloor x \rfloor = 0$, which will happen if and only if x is rational. If x is irrational, this expansion continues forever but the sequence of convergents quickly converges to x. As an example of CFE, consider the case where $x = \pi$.

$$\begin{aligned}
\pi &= 3.14\ldots \\
\pi &= 3 + 0.14\cdots \Rightarrow \frac{p_0}{q_0} = 3 \\
\pi &= 3 + \frac{1}{1/0.14\ldots} \\
\pi &= 3 + \frac{1}{7 + 0.06\ldots} \Rightarrow \frac{p_1}{q_1} = 3 + \frac{1}{7} = \frac{22}{7}
\end{aligned}$$

1.1.1 Properties of Continued Fraction

Recall from equation (6) that q_k is the denominator of the kth convergent. It will be always be true that

$$q_{k+1} \geq 2q_k \tag{8}$$

and

$$\left| \frac{p_k}{q_k} - x \right| \leq \frac{1}{q_k^2}. \tag{9}$$

From these two equations, it should be clear that CFE converges very quickly.

Additionally, if

$$\left| \frac{p}{q} - x \right| \leq \frac{1}{2q_k^2} \tag{10}$$

and $\gcd(p, q) = 1$ then p/q appears as a convergent for some iteration of the CFE of x. Note the similarity between equation (10) and equation (3). If we set $N = 2M^2$ and perform the order finding procedure to get some $\tilde{\omega}_j$, we can use CFE on $\tilde{\omega}_j$ to recover j and r. It follows from equation (8) that the number of convergents we need to calculate is logarithmic in the size of M.

1.2 Fact 2

If we pick two integers, j_1 and j_2, independently and uniformly at random from $\{1, 2, ...r\}$ then

$$\Pr\left[\gcd\left(j_1, j_2\right) = 1\right] \geq 1 - \sum_{p \in \text{prime}}^{r} \frac{1}{p^2} > 1 - \sum_{p \in \text{prime}}^{\infty} \frac{1}{p^2} \geq 0.54. \tag{11}$$

To show the inequality, consider that for any j we pick, the odds that it is divisible by some prime p is asymptotically $1/p$, but will always be $\leq 1/p$. Since j_1 and j_2 are picked independently, the

chance that they would both be divisible by prime p is $\leq 1/p^2$. If we sum over all primes, we get the probability that they share *any* prime factors, thus the inequality shown in equation (11).

In the case that j_1 and j_2 are relatively prime, then $r = \text{lcm}(r'_1, r'_2)$. Since eigenvalue estimation produces j_1 and j_2 that are relatively prime, any factors of r that were canceled in the fraction j'_1/r'_1 could not have been canceled in the fraction j'_2/r'_2. Thus $r = \text{lcm}(r'_1, r'_2)$.

1.3 Quantum Algorithm

We can now describe our order finding algorithm. We first run the eigenvalue estimation twice and get a $\tilde{\omega}_1$, and $\tilde{\omega}_2$. Using CRE, we can determine r'_1 and r'_2 and compute $r = \text{lcm}(r'_1, r'_2)$. Using modular exponentiation, we check whether $a^r \equiv 1 \bmod M$ in time polylog(M). If so, we know that r equals the order of a modulo M, or is a nontrivial multiple. The probability of the former is at least the probability that we have success in equation (1) for both independent runs, and success in equation (11). This puts the total probability of success above 0.35 provided $N \geq 2M^2$. With this probability, we can simply repeat this algorithm several times and output the smallest r retained. This gives the correct result with very high probability.

Let us consider what the total running time of this algorithm is. We naturally choose $N = 2M^2$, so the running time is in terms of M, and using the naive implementation of multiplication, this runs in $\mathcal{O}\left((\log M)^3\right)$. The most efficient known algorithm runs in time $\mathcal{O}\left((\log M)^2(\log\log M)(\log\log\log M)\right)$.

2 Factoring Integers

When asked to factor some number M, we should first check if it is a prime or a prime power. This check can be done in polynomial time, and if M is a prime or prime power, we are done. If M is composite, we need only a means to find a single nontrivial factor, as we can simply divide M by this factor and repeat our factoring algorithm as needed. To find a nontrivial factor, we use two lemmas.

2.1 Necessary Lemmas

The first lemma lets us factor M if we can find some x such that $x^2 \equiv 1 \mod M$ and $x \not\equiv \pm 1 \mod M$.

Lemma 1. *For any integers $x, M > 0$ such that $x^2 \equiv 1 \mod M$ and $x \not\equiv \pm 1 \mod M$, then $\gcd(x \pm 1, M)$ is a nontrivial factor of M.*

Proof. Since

$$x^2 \equiv 1 \mod M, \tag{12}$$

this implies that

$$x^2 - 1 \equiv 0 \mod M. \tag{13}$$

Factoring the left side gives

$$(x-1)(x+1) \equiv 0 \mod M. \tag{14}$$

Since M is divisible by $(x-1)$ and $(x+1)$, clearly $\gcd(x \pm 1, M) \neq 1$. Furthermore we know that $x \not\equiv \pm 1 \mod M$, so M does not divide $x \pm 1$. Therefore there is at least one factor of M that is not in $(x-1)$, and at least one that is not in $(x+1)$. Thus $\gcd(x \pm 1, M)$ is some nontrivial-factor of M. □

The second lemma, combined with our order-finding algorithm, lets us find the x that we use in lemma 1.

Lemma 2. *If M has k distinct prime factors then the probability that $order_M(y)$ is even and that $y^{order(y)/2} \not\equiv \pm 1 \bmod M$ is at least $1 - 1/2^{k-1}$, where y is picked uniformly at random from the set of integers modulo M that are relatively prime to M.*

Proof. We omit the proof, but it uses the Chinese remainder theorem. The full proof can be found in appendix four of [1]. □

2.2 Factoring Algorithm

We can now describe our factoring algorithm for an integer M with k distinct prime factors. If $k = 1$, then M must be either prime or a prime power. However, checking that M is prime or prime power can be done in polynomial time. If M is composite, we pick $y \in \{1, 2, \ldots, M-1\}$ uniformly at random.

If $\gcd(y, M) \neq 1$, then we are done, as the GCD is the nontrivial factor. Otherwise, we check that $\text{order}_M(y)$ is even, and if so, compute $\gcd(y^{\text{order}_M y/2} + 1, M)$ and see if this is a non-trivial factor of M. If so, we are done. Otherwise, we pick another y and try again.

This algorithm can only fail if the $\text{order}_M(y)$ is not even or $y^{order(y)/2} \not\equiv \pm 1 \mod M$, which occurs only with probability of $1/2^{k-1}$. As k is at least 2, the probability of failure is at most $1/2$.

3 Breaking RSA

Besides purely academic interest in factoring numbers, there are also some applications for this algorithm, particularly in cryptographic systems. The best known is breaking RSA public key system, which is widely used in electronic commerce protocols.

If Bob wants to communicate with Alice using the RSA system, Alice will generate two keys. The first is a public key, which she will publish and Bob will use to send messages. The second is a private key which Alice shares with no one.

3.1 Construction

The private key consists of two distinct primes p and q and an integer d such that

$$\gcd(d, (p-1)(q-1)) = 1. \tag{15}$$

p and q are typically chosen at random in a manner that an eavesdropper would have difficulty guessing. Equation (15) implies the existence of integer e such that $de \equiv 1 \mod ((p-1)(q-1))$. Alice can classically compute e efficiently using Euclid's algorithm.

The public key that Alice publishes consists of simply e and n, where $n = pq$.

3.2 Encryption

Suppose that Bob wants to send a message M to Alice where $M \in \{0, 1, \ldots, n-1\}$, but is concerned somebody might eavesdrop on the communication channel and discover M. Instead, Bob will compute cyphertext

$$C = M^e \mod n \tag{16}$$

and send C to Alice.

Alice then computes

$$C^d \mod n \equiv (M^e)^d \mod n \tag{17}$$
$$\equiv M^{1+k(p-1)(q-1)} \mod n \tag{18}$$

Recall Fermat's little theorem, which states that if a prime p and an integer a is coprime with p, then

$$a^{p-1} \equiv 1 \mod p. \tag{19}$$

Additionally, note that if $\gcd(p, q) = 1$ and $n = pq$, then

$$a \equiv b \mod n \Leftrightarrow a \equiv b \mod p \text{ and } a \equiv b \mod q. \tag{20}$$

From these two equations, we can simplify equation (18) to

$$C^d \mod n \equiv M \mod n \tag{21}$$
$$= M. \tag{22}$$

Going from equation (21) to (22) is trivially true as $M < n$.

To illustrate RSA consider the following figure, which shows Bob sending a message to Alice, but there is an eavesdropper listening in on the communication channel. Bob encrypts his message M as C using equation (16), and sends it through the channel as shown in the figure. At this

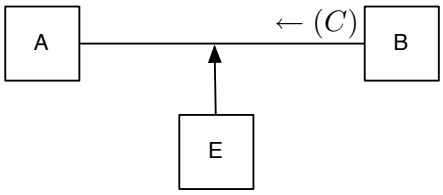

Figure 1: Bob transmits C back to Alice. Alice and Eve both receive it.

point Alice can recover M using equation (22). Eve, having access to C, e and n, can in principle recover M, but the only known ways to do so involve factoring n into p and q, for which no efficient classical algorithms is known.

However, if Eve has a quantum computer, she *can* efficiently factor n, and thus recover M. Therefore, efficient factoring breaks RSA, as we can factor n into p and q. From p and q, we can use e to compute d giving us Alice's private key.

This has far reaching implications, as a malicious party with a sufficiently powerful quantum computer can break many modern electronic commerce and email encryption systems.

References

[1] Michael A. Nielsen and Isaac L. Chuang. Quantum Computation and Quantum Information. *Cambridge*, 2000.