QUANTUM INFORMATION & COMPUTATION

Nilanjana Datta, DAMTP Cambridge

1 Shor's quantum factoring algorithm

We will now describe Shor's quantum factoring algorithm. Given an integer N with $n = \log N$ digits this algorithm will output a factor $1 < K < N$ (or output N if N is a prime) with any chosen constant level of probability $1 - \epsilon$, and the algorithm will run in polynomial time $O(n^3)$. Currently the best known classical algorithm (the so-called number field sieve algorithm) runs in time $e^{O(n^{1/3}(\log n)^{2/3})}$ i.e. there is no known polynomial time classical algorithm for this task.

We'll begin by first describing some pure mathematics (number theory) – involving no quantum ingredients at all – showing how to convert the problem of factoring N into a problem of periodicity determination. Then we'll use our quantum period finding algorithm to achieve the task of factorisation. We'll encounter (and deal with) a technical complication: our function will be periodic on the infinite set $\mathbb{Z}$ of all integers so for computational purposes we need to truncate this down to a *finite* size $\mathbb{Z}_M$ for some M (suitably large, depending on N). Since we do not know the period at the outset the restricted function will not be *exactly* periodic on $\mathbb{Z}_M$: the "last" period will generally be incomplete (as M is not generally an exact multiple of the period). But we'll see that if M is sufficiently large (in fact $M = O(N^2)$ will suffice) then there will be enough complete periods so that the single "corrupted" period has only a negligible effect on our period finding algorithm. We will also always choose M to be a power of 2 to be able to use our explicit circuit for QFT mod M for such M's.

1.1 Factoring as a periodicity problem – some number theory

Let N with $n = \log N$ digits denote the integer that we wish to factorise. We start by choosing $1 < a < N$ at random. Next using Euclid's algorithm (which is a poly-time algorithm) we compute the greatest common divisor $b = gcd(a, N)$. If $b > 1$ we are finished. Thus suppose $b = 1$ i.e. a and N are coprime. We will use:

Theorem 1 *(Euler's theorem): If a and N are coprime then there is a least power $1 < r < N$ such that $a^r \equiv 1 \text{ mod } N$. r is called the order of a mod N.*

We omit the proof which may be found in most standard texts on number theory.

Now consider the powers of a as a function of the index i.e. the modular exponential function:

$$f : \mathbb{Z} \to \mathbb{Z}_N \qquad f(k) = a^k \text{ mod } N \tag{1}$$

Clearly $f(k_1 + k_2) = f(k_1)f(k_2)$ and by Euler's theorem $f(r) = 1$ so $f(k + r) = f(k)$ for all k i.e. f is periodic with period r. Also since r is the *least* integer with $f(r) = 1$ we see that f must be one-to-one within each period.

Next *suppose* we can find r. (We will use our quantum period finding algorithm for this). Suppose r comes out to be even. Then

$$a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \bmod N$$

i.e.

$$N \text{ exactly divides the product } (a^{r/2} - 1)(a^{r/2} + 1) \tag{2}$$

(and knowing r we can calculate each of these terms in poly(n) time).

We know N does not divide $a^{r/2} - 1$ (since r was the *least* power x such that $a^x - 1$ is divisible by N). Thus *if* N does not divide $a^{r/2} + 1$ i.e. if $a^{r/2} \not\equiv -1 \bmod N$, then in eq. (2) N must partly divide into $a^{r/2} - 1$ and partly into $a^{r/2} + 1$. Hence using Euclid's algorithm again, we compute $gcd(a^{r/2} \pm 1, N)$ which will be factors of N.

All this works *provided* r is even and $a^{r/2} \not\equiv -1 \bmod N$. How likely is this, given that a was chosen at random? We quote the following theorem.

Theorem 2 *Suppose N is odd and not a power of a prime. If $a < N$ is chosen uniformly at random with $gcd(a, N) = 1$ then Prob(r is even and $a^{r/2} \not\equiv -1 \bmod N$) $\geq 1/2$.*

For a proof of this result see Preskill's notes page 307 et seq., Nielsen/Chuang appendix 4.3 or A. Ekert and R. Jozsa, *Reviews of Modern Physics*, vol 68, p733-753 1996, appendix B.

Hence for any N which is odd and not a prime power, we will obtain a factor with probability at least half. Given any candidate factor we can check it (in poly(n) time) by test division into N. Thus repeating the process, say 10 times, we will fail to get a factor only with tiny probability $1/2^{10}$, and succeed with any probability $1 - \epsilon$ with $\log_2 1/\epsilon$ repetitions.

Example 1 *Consider $N = 15$ and choose $a = 7$. Then a direct calculation shows that the function $f(k) = 7^k \bmod 15$ for $k = 0, 1, 2, \ldots$ has values 1,7,4,13,1,7,4,13,... so $r = 4$. Thus $7^4 - 1 = (7^2 - 1)(7^2 + 1) = (48)(50)$ is divisible by 15 and computing $gcd(15, 48) = 3$ and $gcd(15, 50) = 5$ gives non-trivial factors of 15.*

All of this works if N is not even or a prime power. So how do we recognise and treat these latter cases? If N is even (which is easy to recognise!) we immediately have a factor 2 and we are finished. If $N = p^l$ is a prime power then we can identify this case and find p using the following result (which we quote without proof).

Lemma 1 *Suppose $N = c^l$ for some integers $c, l \geq 2$. Then there is a classical polynomial time algorithm that outputs c.*

Running this algorithm on *any* N will output *some* number c' and we can check if it divides N or not. If N was a prime power p^l then c' will be p.

Summarizing the process so far: given N we proceed as follows.
(i) Is N even? If so, output 2 and stop.
(ii) Run the algorithm of lemma 1, test divide the output and stop if a factor of N is obtained.
(iii) If N is neither even nor a prime power choose $1 < a < N$ at random and compute $s = gcd(a, N)$. If $s \neq 1$ output s and stop.
(iv) If $s = 1$ find the period r of $f(k) = a^k \bmod N$. (We will achieve this with any desired level of constant probability $1 - \epsilon$ using the quantum algorithm described in the next section).
(v) If r is odd, go back to (iii). If r is even compute $t = gcd(a^{r/2} + 1, N)$, so by definition t is a factor of N. If $t \neq 1, N$ output t. If $t = 1$ or N go back to (iii) and try again.

According to theorem 2 any run of (iv) and (v) will output a factor with probability $> 1/2$ so K repetitions of looping back to (iii) will all fail only with probability $< 1/2^K$ which can be made as small as we like.

1.2 Computing the period of $f(k) = a^k \bmod N$

Let r denote the (as yet unknown) period of $f(k) = a^k \bmod N$ on the infinite domain $\mathbb{Z}$. We will work on the finite domain $D = \{0, 1, \ldots, 2^m - 1\}$ where 2^m is the least power of 2 greater than N^2 (see later for the reason for this choice). Let $2^m = Br + b$ with $1 < b < r$ i.e. the domain D contains B full periods and only the initial part up to b of the next period. Using a standard application of computation by quantum parallelism we manufacture the state $\frac{1}{\sqrt{2^m}} \sum_{x \in D} |x\rangle \, |f(x)\rangle$ and measure the second register to obtain some value $y_0 = f(x_0)$ with $0 \leq x_0 < r$. In the first register we get the state

$$|per\rangle = \frac{1}{\sqrt{A}} \sum_{k=0}^{A-1} |x_0 + kr\rangle$$

where

$$A = \left\{ \begin{array}{ccll} B+1 & = & \lfloor \frac{2^m}{r} \rfloor + 1 & \text{if } x_0 < b \\ B & = & \lfloor \frac{2^m}{r} \rfloor & \text{if } x_0 \geq b. \end{array} \right. \tag{3}$$

Let

$$QFT_{2^m} \, |per\rangle = \sum_{c=0}^{2^m - 1} g(c) \, |c\rangle \, .$$

Writing $\omega = e^{2\pi i/2^m}$ we have

$$g(c) = \frac{1}{\sqrt{A}\sqrt{2^m}} \sum_{k=0}^{A-1} \omega^{c(x_0 + kr)} = \frac{\omega^{cx_0}}{\sqrt{A}\sqrt{2^m}} \left[\sum_{k=0}^{A-1} \omega^{crk} \right] .$$

As before (in Lecture Note 10, where c was called y) the square bracket is a geometric series with ratio $\alpha = \omega^{cr}$ and we have

$$[\ldots] = 1 + \alpha + \alpha^2 + \ldots + \alpha^{A-1} = \begin{cases} \frac{1-\alpha^A}{1-\alpha} & \text{for } \alpha \neq 1 \\ A & \text{for } \alpha = 1. \end{cases}$$

Let's look more closely at the ratio $\alpha = e^{2\pi i c r / 2^m}$. Previously we had r dividing the denominator 2^m exactly and $2^m/r = A$ so if $\alpha \neq 1$ then α was an A^{th} root of unity and the geometric series summed to *zero* in all these cases. The only c values that survived were the exact multiples of $A = 2^m/r$ having $\alpha = 1$. There were r such multiples each with equal |amplitude| of $\frac{1}{\sqrt{r}}$.

In the present case r does not divide 2^m exactly generally so α is not an A^{th} root of unity and we do not get a lot of "exactly zero" amplitudes for $|c\rangle$'s! However we aim to show that a measurement on QFT$|per\rangle$ will yield an integer c-value which is *close* to a multiple of $2^m/r$ with suitably high probability.

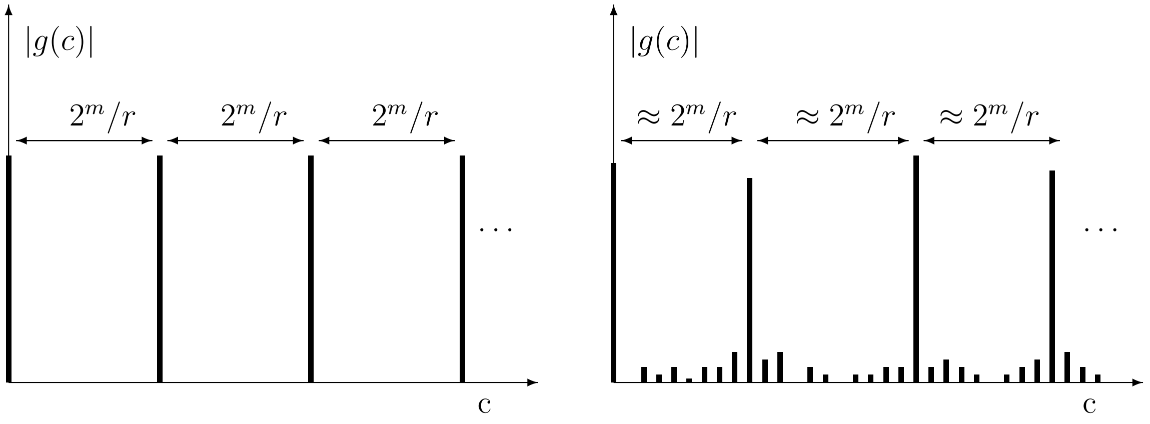
Consider the r multiples of $2^m/r$ (which are now not integers necessarily!):

$$0, \frac{2^m}{r}, 2(\frac{2^m}{r}), \ldots, (r-1)(\frac{2^m}{r}).$$

Each of these is within half of a unique nearest integer. Note that $k(2^m/r)$ can never be exactly half way between two integers since $r < N$ and $2^m > N^2$, so (using 2's in 2^m) all factors of 2 can be cancelled out of the denominator r. Thus we consider c values (r of them) such that

$$|c - k\frac{2^m}{r}| < \frac{1}{2} \qquad k = 0, 1, \ldots, (r-1). \tag{4}$$

In the previous case of exact periodicity (where $2^m/r$ was an integer) each of these c-values appeared with probability $1/r$ and all other c-values had probability zero. Here we will show that although the other c-values will generally have non-zero probabilities, the special ones in eq. (4) still have probability at least γ/r for a constant γ.



(a) exact periodicity

(b) inexact periodicity

Figure 1.2: Schematic depiction of amplitudes in QFT$|per\rangle$. (a) exact periodicity (r divides 2^m): we have nonzero amplitudes only at exact multiples $c = k2^m/r$. (b) non-exact periodicity: we have nonzero amplitudes for many c-values but the integers nearest to the multiples $k2^m/r$ still have suitably large amplitudes.

Theorem 3 *Suppose we measure the label in QFT$|per\rangle$. Let c_k be the unique integer with $|c - k\frac{2^m}{r}| < \frac{1}{2}$. Then prob$(c_k) > \gamma/r$ where $\gamma \approx 4/\pi^2$.*

Proof: (optional) For any c we have

$$\text{prob}(c) = |g(c)|^2 = \frac{1}{A2^m}\left|\frac{1-\alpha^A}{1-\alpha}\right|^2$$

with $\alpha = e^{2\pi i cr/2^m} = e^{2\pi i (cr \bmod 2^m)/2^m}$. For our special c-values satisfying eq. (4) we have $|cr - k2^m| < r/2$ so

$$-\frac{r}{2} < cr \bmod 2^m < \frac{r}{2}. \tag{5}$$

Write $\alpha = e^{i\theta_c}$ with $\theta_c = 2\pi(cr \bmod 2^m)/2^m$ so $|\theta_c| < \pi r/2^m$. Also from eq. (3) we see that in all cases $A < 2^m/r + 1$ so

$$|A\theta_c| < \frac{\pi r}{2^m}A < \pi(1 + \frac{r}{2^m}).$$

Write $A\theta_{max} = \pi(1 + r/2^m)$. Note that for all c

$$0 \leq |A\theta_c/2| < A\theta_{max}/2 < \pi. \tag{6}$$

To estimate prob(c) we'll use the algebraic identity

$$\left|\frac{1-e^{iA\theta}}{1-e^{i\theta}}\right|^2 = \left(\frac{\sin A\theta/2}{\sin\theta/2}\right)^2.$$

We have

$$\begin{aligned}
\text{Prob}(c) &= \tfrac{1}{A2^m}\left(\tfrac{\sin A\theta_c/2}{\sin\theta_c/2}\right)^2 \\
&> \tfrac{1}{A2^m}\left(\tfrac{\sin A\theta_c/2}{\theta_c/2}\right)^2 \quad (\text{as } \sin x < x) \\
&= \tfrac{A}{2^m}\left(\tfrac{\sin A\theta_c/2}{A\theta_c/2}\right)^2 \\
&> \tfrac{A}{2^m}\left(\tfrac{\sin A\theta_{\max}/2}{A\theta_{\max}/2}\right)^2
\end{aligned}$$

where the last inequality follows from eq. (6) and the fact that $\frac{\sin x}{x}$ is decreasing on $0 < x < \pi$.

Next from eq. (3) we have $A > 2^m/r - 1$ so $\frac{A}{2^m} > \frac{1}{r} - \frac{1}{2^m}$. Introducing $g(x) = \left(\frac{\sin x}{x}\right)^2$ we have

$$\text{prob}(c) > (\frac{1}{r} - \frac{1}{2^m})g(A\theta_{max}/2) = \frac{1}{r}(1 - \frac{r}{2^m})g(A\theta_{max}/2) > \frac{\gamma}{r} \tag{7}$$

for a constant γ, noting that $2^m > N^2$ and $r < N$ so $r/2^m << 1$ for all large N. To get a proper lower bound for γ is straightforward but a little messy. Here we will just consider the case of very large N and ignore terms of order $r/2^m < 1/N$. We have $A\theta_{max}/2 = \frac{\pi}{2}(1+r/2^m) \approx \pi/2$ so $g(\pi/2) = (2/\pi)^2$ and from eq. (7) we get $\text{prob}(c) > \gamma/r$ for $\gamma \approx 4/\pi^2$. □

According to this theorem, for each $k = 0, \ldots, r-1$ we will obtain the unique c-value satisfying eq. (4) with probability at least γ/r. We will be especially interested in those c's for which the corresponding k is *coprime* to r and there are $O(r/\log\log r)$ of these. Hence the total probability of obtaining such a "good" c-value is $O(1/\log\log r) > O(1/\log\log N)$ and with $O(\log\log N)$ repetitions we will obtain such a good c-value with any desired constant level of probability. To complete the determination of r and hence the description of the quantum factoring algorithm, it remains to show that r can be determined from a ("good") c-value in time $\text{poly}(\log N)$.

1.3 Getting r from a good c value

Suppose we have c satisfying eq. (4) i.e.

$$\left| \frac{c}{2^m} - \frac{k}{r} \right| < \frac{1}{2^{m+1}}. \tag{8}$$

Recall that $r < N$ and $2^m > N^2$ so

$$\left| \frac{c}{2^m} - \frac{k}{r} \right| < \frac{1}{2N^2} \quad \text{with } r < N \tag{9}$$

and $c/2^m$ is a *known* fraction. We claim that there is at most one fraction k'/r' with a denominator r' less than N satisfying eq. (9). Hence for given $c/2^m$, eq. (9) determines k/r uniquely. To prove this claim suppose k'/r' and k''/r'' both lie within $1/(2N^2)$ of $c/2^m$. Then

$$\left| \frac{k'}{r'} - \frac{k''}{r''} \right| = \frac{|k'r'' - r'k''|}{r'r''} \geq \frac{1}{r'r''} > \frac{1}{N^2} \tag{10}$$

But k'/r' and k''/r'' are both within $1/(2N^2)$ of $c/2^m$ so they must be within $1/N^2$ of each other, contradicting eq. (10). Hence there is at most one k/r with $r < N$ satisfying eq. (9).

This result is the reason why we chose 2^m to be greater than N^2: it guarantees that the bound on RHS of eq. (9) is $< 1/(2N^2)$ and then k/r is uniquely determined from $c/2^m$.

Example 2 *Suppose we wish to factor* $N = 39$ *and we have chosen* $a = 7$ *which is coprime to* N. *Let* r *be the period of* $f(x) = 7^x \bmod 39$. *We have* $N^2 = 1521$ *and* $2^{10} < N^2 < 2^{11} = 2048 = 2^m$ *so* $m = 11$. *Suppose the measurement of* $QFT_{2^m}\left|per\right\rangle$ *yields* $c = 853$. *According to our theory, this number has a "reasonable" probability to be within half of a multiple* $k2^{11}/r$ *of* $2^m/r$. *If this is actually the case then our theory guarantees that the fraction* k/r *is uniquely determined, as the unique fraction* k/r *with denominator* < 39 *that is within* $1/2^{m+1} = 1/2^{12}$ *of* $853/2048$. *In this example we can* *(with a calculator) check all fractions a/b with $a < b < N = 39$ to see which ones (if any) satisfy*

$$\left|\frac{a}{b} - \frac{853}{2048}\right| < \frac{1}{2^{12}}. \tag{11}$$

There are $O(N^2)$ such fractions to try. We find that there is only one viz. $a/b = 5/12$ that satisfies eq. (11):

$$\left|\frac{a}{b} - \frac{853}{2048}\right| = 0.000163 < \frac{1}{2^{12}} = 0.000244$$

This result is consistent with $k = 5$ and $r = 12$ and also with $k = 10$ and $r = 24$. But our theory also guarantees that k is coprime to r with "reasonable" probability which in this case sets $r = 12$. We can then verify that 7^{12} is indeed congruent to 1 mod 39 and 7^x for all $x < 12$ is not congruent to 1 so $r = 12$ is the correct period.

So far we have that k/r is *uniquely determined* by $c/2^m$ but how do we actually compute k/r from $c/2^m$? In the above example we were able to try out all candidate fractions k'/r' with denominator less than N. But there are generally $O(N^2)$ such fractions to try so this method of seeking the unique one is *not efficient*, requiring at least $O(N^2)$ steps, which is exponential in $n = \log N$!

To obtain an efficient (i.e. poly(n) time) method we invoke the elegant mathematical:

Theory of continued fractions

Any rational number s/t (with $s < t$) may be expressed as a so-called continued fraction (CF):

$$\frac{s}{t} = \cfrac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\cdots + \frac{1}{a_l}}}} \tag{12}$$

where $a_1, \ldots, a_l$ are positive integers. To do this we begin by writing $s/t = 1/(t/s)$. Since $s < t$ we have $t/s = a_1 + s_1/t_1$ with $a_1 \geq 1$ and $s_1 < t_1 = s$ and so

$$\frac{s}{t} = \frac{1}{a_1 + \frac{s_1}{t_1}}.$$

Then repeating with s_1/t_1 we get $t_1/s_1 = a_2 + s_2/t_2$, $t_2 = s_1$ and

$$\frac{s}{t} = \frac{1}{a_1 + \frac{1}{a_2 + \frac{s_2}{t_2}}}.$$

Continuing in this way we get a sequence of integers a_k, s_k and t_k. Note that $s_k < t_k$ and t_{k+1} is always given by s_k. Hence the sequence t_k of denominators is strictly a decreasing sequence of non-negative integers and hence the process must always terminate, after some number l, of iterations giving the expression in eq. (12).

To avoid the cumbersome "fractions of fractions" notation in eq. (12) we will write

$$\cfrac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\cdots + \frac{1}{a_l}}}} = [a_1, a_2, \ldots, a_l]. \tag{13}$$

For each $k = 1, \ldots, l$ we can truncate the fraction in (13) at the $k^{\rm th}$ level to get a sequence of rational numbers

$$\frac{p_1}{q_1} = [a_1] = \frac{1}{a_1}, \quad \frac{p_2}{q_2} = [a_1, a_2] = \frac{1}{a_1 + \frac{1}{a_2}} = \frac{a_2}{a_1 a_2 + 1}, \quad \ldots$$

$$\frac{p_k}{q_k} = [a_1, \ldots, a_k], \quad \ldots \quad \frac{p_l}{q_l} = [a_1, \ldots, a_l] = \frac{s}{t}.$$

p_k/q_k is called the $k^{\rm th}$ *convergent* of the continued fraction of s/t.

Continued fractions enjoy the following tantalising properties.

Lemma 2 *Let $a_1, \ldots, a_l$ be any positive numbers (not necessarily integers here). Set $p_0 = 0$, $q_0 = 1$, $p_1 = 1$ and $q_1 = a_1$.*
(a) Then $[a_1, \ldots, a_k] = p_k/q_k$ where

$$p_k = a_k p_{k-1} + p_{k-2} \qquad q_k = a_k q_{k-1} + q_{k-2} \qquad k \geq 2. \tag{14}$$

Note that if the a_k's are integers then so are the p_k's and q_k's.
(b) $q_k p_{k-1} - p_k q_{k-1} = (-1)^k$ for $k \geq 1$.
(c) If $a_1, \ldots, a_l$ are integers then gcd$(p_k, q_k) = 1$ for $k \geq 1$.

Proof outline (optional):
(a) By induction on k. For the base case $k = 2$ direct calculation gives $[a_1, a_2] = a_2/(a_1 a_2 + 1)$ and eq. (14) correctly gives $p_2 = a_2$ and $q_2 = a_1 a_2 + 1$. Thus suppose eq. (14) holds for length k. For length $k+1$ we have $[a_1, \ldots a_k, a_{k+1}] = [a_1, \ldots, a_{k-1}, a_k + 1/a_{k+1}]$ where the RHS now has length k. Let $\tilde{p}_j/\tilde{q}_j$ be the sequence of convergents of RHS. Then $\tilde{p}_k/\tilde{q}_k = [a_1, \ldots a_k, a_{k+1}] = [a_1, \ldots, a_{k-1}, a_k + 1/a_{k+1}]$ and clearly $\tilde{p}_{k-1} = p_{k-1}$, $\tilde{p}_{k-2} = p_{k-2}$ and similarly for the q's. Hence using the recurrence relation eq. (14) at length k (twice) we get:

$$\frac{\tilde{p}_k}{\tilde{q}_k} = \frac{(a_k + 1/a_{k+1})p_{k-1} + p_{k-2}}{a_k + 1/a_{k+1})q_{k-1} + q_{k-2}} = \frac{p_k + p_{k-1}/a_{k+1}}{q_k + q_{k-1}/a_{k+1}} = \frac{a_{k+1}p_k + p_{k-1}}{a_{k+1}q_k + q_{k-1}}$$

i.e. eq. (14) holds for $k+1$.

(b) is proved by induction on k using the recurrence relations of (a) to express the $(k, k-1)$ expression in terms of the same expression with lower values of the subscripts.

(c) follows from (b): if a divides p_k and q_k exactly then by (b), a must divide ±1 i.e. $a = 1$. □

Theorem 4 *Consider the continued fraction $s/t = [a_1, \ldots, a_l]$. Let $p_k/q_k = [a_1, \ldots, a_k]$ be the $k^{\rm th}$ convergent for $k = 1, \ldots, l$. If s and t (cancelled to lowest terms) are m bit integers then the length l of the continued fraction is $O(m)$ and this continued fraction together with its convergents can be calculated in time $O(m^3)$.*

Proof outline (optional):
We have $a_k \geq 1$ and $p_k, q_k \geq 1$ so by the above recurrence relations, p_k and q_k must be increasing sequences and $p_k = a_k p_{k-1} + p_{k-2} \geq 2p_{k-2}$. Similarly $q_k \geq 2q_{k-2}$. Hence p_k and q_k are each $\geq 2^{\lfloor k/2 \rfloor}$ so since p_k and q_k are coprime and increasing, we must get s/t after at most $l = O(m)$ iterations. The computation of each successive a_k involves the division of $O(m)$ bit integers (and splitting off the integer parts). These arithmetic operations can be performed in $O(m^2)$ time so we can compute all $O(m)$ a_k's in $O(m^3)$ time. Similarly using the recurrence relation we can compute all p_k's and q_k's in $O(m^3)$ time too. □

Theorem 5 *Let $0 < x < 1$ be a rational number and suppose that p/q is a rational number such that*

$$\left| x - \frac{p}{q} \right| < \frac{1}{2q^2}.$$

Then p/q is a convergent of the continued fraction of x.

Proof (optional):
Let $p/q = [a_1, \ldots, a_n]$ be the CF expansion of p/q with convergents p_j/q_j, so $p_n/q_n = p/q$. Introduce δ defined by

$$x = \frac{p_n}{q_n} + \frac{\delta}{2q_n^2} \tag{15}$$

so $|\delta| < 1$. We aim to show that the CF of x is an extension of the CF of p/q i.e. we want to construct λ rational so that $x = [a_1, \ldots, a_n, \lambda]$. In view of lemma 2(a) define λ by $x = (\lambda p_n + p_{n-1})/(\lambda q_n + q_{n-1})$. Using eq. (15) to replace x we get

$$\lambda = 2\left(\frac{q_n p_{n-1} - p_n q_{n-1}}{\delta}\right) - \frac{q_{n-1}}{q_n}.$$

By lemma 2(b), $q_n p_{n-1} - p_n q_{n-1} = (-1)^n$. We may assume that this is the same as the sign of δ since if it is the opposite sign then from the start write $p/q = [a_1, \ldots, a_n - 1, 1]$ so the value of n is increased by 1 and the sign is flipped. Thus without loss of generality we can assume that $(q_n p_{n-1} - p_n q_{n-1})/\delta$ is positive and so

$$\lambda = \frac{2}{\delta} - \frac{q_{n-1}}{q_n} > 2 - 1 > 1$$

(as $|\delta| < 1$ and $q_{n-1} < q_n$). Next let $\lambda = b_0 + \lambda'$ where b_0 is te integer part and $0 < \lambda' < 1$ and write $\lambda' = [b_1, \ldots, b_m]$. So $x = [a_1, \ldots, a_n, \lambda] = [a_1, \ldots, a_n, b_0, b_1, \ldots, b_m]$ i.e. p/q is a convergent of the CF of x as required. (In the last argument we also used the easily proven fact that the CF expansion of any number is unique, except for the above trick of splitting 1 off from the last term i.e. if $[a_1, \ldots, a_n] = [b_1, \ldots, b_m]$ and $a_n, b_m \neq 1$ then $m = n$ and $a_i = b_i$). □

Remark: Theorem 5 actually remains true for irrational x too. For an irrational number the continued fraction development does not terminate – we get an infinitely long continued fraction and corresponding infinite sequence of rational convergents p_k/q_k $k = 1, 2, \ldots$. This sequence provides an efficient method of computing excellent rational approximations to an irrational recalling that q_k grows exponentially with k and (by theorem 5) it determines the accuracy of the approximation. $\Box$

Now let us return to our problem of getting r from the knowledge of c and 2^m satisfying eq. (9):

$$\left| \frac{c}{2^m} - \frac{k}{r} \right| < \frac{1}{2N^2} \quad \text{and } r < N.$$

We know that there is (at most) a unique such fraction k/r and according to theorem 5 this fraction must be a convergent of the continued fraction of $c/2^m$. Since $2^m = O(N^2)$ we have that c and 2^m are $O(n)$ bit integers and the computation of all the convergents can be performed in time $O(n^3)$. So we do this computation and finally check through the list of $O(n)$ convergents to find the unique one satisfying eq. (9), and read off r as its denominator.

Example 3 *(Continuation of example 2).*
Suppose we have obtained $c = 853$ *with* $2^m = 2^{11} = 2048$*. We develop* $853/2048$ *as a continued fraction:*

$$\frac{853}{2048} = 1/(2048/853); \quad \frac{2048}{853} = 2 + \frac{342}{853}; \quad \frac{853}{243} = 2 + \frac{169}{342};$$

$$\frac{342}{169} = 2 + \frac{4}{169}; \quad \frac{169}{4} = 42 + \frac{1}{4}; \quad \frac{4}{1} = 4 + 0$$

so

$$\frac{853}{2048} = [2, 2, 2, 42, 4].$$

The convergents are

$$[2] = \frac{1}{2}; \quad [2,2] = \frac{2}{5}; \quad [2,2,2] = \frac{5}{12}; \quad [2,2,2,42] = \frac{212}{509}; \quad [2,2,2,42,4] = \frac{852}{2048}.$$

Checking these five fractions we find only $5/12$ *as being within* $1/2^{12}$ *of* $853/2048$ *and having denominator* < 39*.*

In appendix 1.4 we will reconsider all the ingredients of Shor's quantum factoring algorithm and assess its polynomial time complexity in more detail.

1.4 Assessing the complexity of Shor's algorithm

This subsection is optional and not examinable.

Let us now consider all the parts of the quantum factoring algorithm and assess the time complexity of the whole process. Recall that the best known classical algorithm to factor N with $n = \log N$ digits runs in a time that's exponential in $n^{1/3}$.

Consider the case where N is neither even nor a prime power and $a < N$ chosen at random is coprime to N. In this case we must proceed to use the quantum part of the overall algorithm summarised at the end of section 1.1 i.e. the quantum part (iv), in addition to some further classical computational steps as well.

We first need to compute the function $f(k) = a^k \mod N$ (in superposition) over a domain $0 \leq k < 2^m$ where $2^m = O(N^2)$ so $m = O(n)$. To compute a^k we use repeated squaring of a $\lfloor \log k \rfloor$ times. Once the exponent is close to k we do a few more multiplications to reach k itself. This requires $O(\log k) = O(m) = O(n)$ multiplications of integers mod N. Each such multiplication can be performed in $O(n^2)$ time (by the standard "long multiplication" algorithm) so the computation of $f(k)$ for any $0 \leq k < 2^m$ can be performed in $O(n^3)$ steps. To compute the uniform superposition of all inputs for this computation we need $m = O(n)$ initial Hadamard operations. Thus the state $|f\rangle = \frac{1}{\sqrt{2^m}} \sum |k\rangle |f(k)\rangle$ can be computed in $O(n^3)$ steps.

Remark
There exist algorithms for integer multiplication that are faster than $O(n^2)$ time, running in time $O(n \log n \log \log n)$ so the above $O(n^3)$ can be improved to $O(n^2 \log n \log \log n)$. □

Next we perform measurements on the output register of $O(n)$ qubits i.e. $O(n)$ single qubit measurements. Then we apply QFT $\mod 2^m$ to obtain the state QFT$|per\rangle$. In Lecture Note 10, it was given that QFT $\mod 2^m$ may be implemented in $O(m^2) = O(n^2)$ steps.

Next we measure the state QFT$|per\rangle$ ($O(n)$ single qubit measurements again) to obtain the value that we called c in section 1.3. Thus to get such a value the number of steps is $O(n^2 \log n \log \log n) + O(n) + O(n^2) + O(n) = O(n^2 \log n \log \log n)$. To get the period r we need c to be a "good" c value i.e. $c/2^m$ is close to a multiple k/r of $1/r$ where k is coprime to r. To achieve this with a constant level of probability, $O(\log \log N) = O(\log n)$ repetitions of the above process suffice i.e. $O(n^2 (\log n)^2 \log \log n)$ steps in all.

Remark
Actually it may be shown that a *constant* number of repetitions suffices here (instead of $O(\log n)$) to determine r. Suppose that in two repetitions we obtain k_1/r and k_2/r with neither k_1 nor k_2 coprime to r. Then we will determine r_1 and r_2 which are the denominators of k_1/r and k_2/r cancelled to lowest terms i.e. r_1 and r_2 will be randomly chosen factors of r. Then, according to a further theorem of number theory, if we compute the least common multiple $\tilde{r}$ of r_1 and r_2 we will have $\tilde{r} = r$ with probability at least 1/4. □

To get r from c we use the (classical) continued fractions algorithm which required $O(n^3)$ steps. Finally to obtain our factor of N we (classically) compute $t = gcd(a^{r/2+1}, N)$ using Euclid's algorithm which requires $O(n^3)$ steps for n digit integers. If r was odd or r is even but $t = 1$ then we go back to the start. But we saw that the good case "r is even and $t \neq 1$" will occur with any fixed constant level of probability $1 - \epsilon$ after a constant number $O(\log 1/\epsilon)$ of such repetitions.

Hence the time complexity of the entire algorithm is $O(n^3)$ (or actually slightly better with optimized algorithms and a more careful complicated analysis). It is amusing to note that the "bottlenecks" of the algorithms performance i.e. the sections requiring the highest degree polynomial running times, are actually the *classical* processing sections and not the novel quantum parts!