Name: **Huan Q. Bui**
Course: **8.370 - QC**
Problem set: **#7**
Due: Wednesday, Nov 9, 2022
Collaborators/References: Prof. Shor's Office Hours, Prof. Shor's factoring paper (1995), Nielsen and Chuang, Elba Alonso-Monsalve, Shira Asa-El

**1. Factoring 21.** Suppose we use a unitary transformation which acts as

$$U\left|u \mod 21\right\rangle = \left|2y \mod 21\right\rangle$$

on binary representation of numbers between 0 and 20. For the factoring algorithm we start with the state $|1 \mod 21\rangle$ and use phase estimation on the unitary $U$. If we have an eigenvector with eigenvalue $e^{2\pi i a/b}$. we will assume that phase estimation combined with continued fraction returns $a/b$ in reduced fraction form with probability 1.

(a) The state $|1 \mod 21\rangle$ can be represented as a superposition of the eigenvectors of $U$. To find what these are, we first have to find the smallest $r > 0$ for which $2^r \equiv 1 \mod 21$.

| $a^r$ | $2^0$ | $2^1$ | $2^2$ | $2^3$ | $2^4$ | $2^5$ | $2^6$ | $2^7$ | $2^8$ |
|---|---|---|---|---|---|---|---|---|---|
| $a^r \mod 21$ | 1 | 2 | 4 | 8 | 16 | 11 | 1 | 2 | 4 |

So $r = 6$. We conclude that there are $\boxed{6}$ eigenvectors of $U$. They are of the form

$$\begin{aligned}\left|\zeta_k\right\rangle = \frac{1}{\sqrt{6}}\Big(&\left|2^0 \mod 21\right\rangle + e^{2\pi i k/6}\left|2^1 \mod 21\right\rangle + e^{4\pi i k/6}\left|2^2 \mod 21\right\rangle \\ &+e^{6\pi i/6}\left|2^3 \mod 21\right\rangle + e^{8\pi i k/6}\left|2^4 \mod 21\right\rangle + e^{10\pi i k/6}\left|2^5 \mod 21\right\rangle\Big), \quad k \in \{0, 1, \ldots, r-1\}\end{aligned}$$

The state $|1 \mod 21\rangle$ is then a superposition of these states $\{\zeta_k\}_{k=0}^{r-1}$:

$$\left|1 \mod 21\right\rangle = \frac{1}{\sqrt{6}}\sum_{k=0}^{r-1}\left|\zeta_k\right\rangle.$$

(b) Suppose the algorithm returns $k'/r'$ (which is in reduced form) and if $r'$ is even, we compute $2^{r'/2}$ and try to use it to factor. What is the probability that we get the right factors this way? To do this, we first list what we have done right so far:

- $g = 2$ is relatively prime to 21, so it generates the cyclic group of order 21. The order of $g = 2$ is $r = 6$.
- $g = 2$ is also a good choice because $g^{r/2} - 1 = 2^{6/2} - 1 = 7 \neq 21$ and $g^{r/2} + 1 = 2^{6/2} + 1 = 9 \neq 21$.

This means that the algorithm fails when $k'/r'$ (in reduced form) cannot be used to find $r$. The possible values that we can get are

$$\frac{k'}{r'} \in \left\{\frac{0}{6}, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}\right\}.$$

These values of $k'/r'$ occur with equal probabilities. Of them, we find $r' = 3$ a third of the time. When $r' = 2$ (which occurs a sixth of the time) or $k = 0$ (which also occurs a sixth of the time) we also reject them. So, we get the right factors $\boxed{1/3}$ of the time this way. We could also accept $r' = 2$ on the basis that since $r'$ divides $r < 21$ we can multiply $r'$ by small factors $n$ and check this result until we get $nr' = 6$. So we could also say that we get the right factors $\boxed{1/2}$ of the time.

**2. Factoring algorithm failing.** In the last problem, we saw that even if the phase estimation and continued fractions pieces of the algorithm work perfectly, and we choose a $g \in [1, N-1]$ that is relatively prime to $N$, use the unitary

$$U : |y \mod N\rangle \to |gy \mod N\rangle$$

and get the correct period r, the factoring algorithm still has some probability of failing.

(a) How bad is this problem for large $N$?

To answer this question suppose that $g$ is chosen so that it is relatively prime to $N$ and has order $r$. Now suppose that the period-finding subroutine gives us $k'/r'$ in irreducible form. If $k' = 0$ then we get no information. If $k' \neq 0$, knowing what $r'$ is, we can check whether $r' = r$ by verifying $(2^{r'/2} + 1)(2^{r'/2} - 1) = N$. Our main concern here is that $r' \neq n$ because $k, n$ are not co-prime. So, we want to ask: about how many times do we have to repeat the process in order to find $k'/r'$ for which $r' = r$?

One way to answer is question is by calculating the probability of find $k \in \{0, 1, \ldots, r-1\}$ for which $\gcd(k, r) = 1$. It is easy to see that

$$\Pr\big(k | \gcd(k, r) = 1\big) = \frac{\phi(r)}{r} \geq \frac{C}{\ln \ln r} \geq \frac{C}{\ln \ln N}$$

since $r \leq N$. So, we are expected to repeat the algorithm $O(\ln \ln N)$ times to get the right answer.

Another method is due to Prof. Shor in his 1995 factoring algorithm paper: We have to find $k'/r'$ for which $\gcd(k', r') = 1$ and therefore $k' = k, r' = r$ and of course $\gcd(k, r) = 1$. Before we proceed, we note that because of this if $r' \neq r$ then $r'$ divides $r$ unless $k = 0$ which occurs with probability $1/r \leq 1/2$, which we can compensate more by iterating the algorithm a few extra times.

Now we proceed. Also, let's not worry about whether $r$ is odd or even for now:

- How many states $|k'/r'\rangle \left|g^k \mod N\right\rangle$ allow us to find $r$ this way? For each $r$, there are $\phi(r)$ values for $k$ for which $k, r$ are co-prime. Here, $\phi(\cdot)$ denotes the Euler's totient function. Moreover, there can only be $r$ unique values for $g^k \mod N$. So, there are $r\phi(r)$ state $|k'/r'\rangle \left|g^k \mod N\right\rangle$ that allows us to find $r$.
- How often do these states occur? They occur with probability of at least $4/(\pi^2 r^2)$. This means that we can obtain $r$ with at least $4\phi(r)/(\pi^2 r)$ probability. Since $\phi(r) \geq Cr/(\ln \ln r)$ for some constant $C$, we conclude that we find $r$ with probability at least $D/\ln \ln r$ for some constant $D$.
- How do we put this in terms of $N$? Since $r \leq N$, so we may conclude that we find $r$ at least a $D/\ln \ln N$ fraction of the time. This means that we may have to repeat the process $\boxed{O(\ln \ln N)}$ times.

Should we worry about the case where $r$ is odd or $g^{r/2} \equiv -1$ and how likely they occur? It turns out that the answer is **No**. This is a consequence of Theorem 5.3 of Nielsen and Chuang, which states that

$$\Pr\left(r \text{ even and } g^{r/2} \neq -1 \mod N\right) \geq 1 - \frac{1}{2^m}$$

which could be made arbitrarily close to 1. Correspondingly, the probability of the other failure modes ($r$ odd or $g^{r/2} \equiv 1$) can be made arbitrarily small. The point is that we don't have to worry about these.

(b) Can you think of anything to do to make the factoring algorithm somewhat more efficient for large N? (Assume that quantum computation is expensive and classical computation is relatively cheap.)

A quick way to get improvement is this: Since we know that $r'$ is a factor of $r$, we can simply check $r'$ times small factors, $r', 2r', 3r', \ldots$ until we get the right answer. For how long do we have to do this? This question is not relevant to the problem, but I think the answer is "not long," i.e., maybe $O(\log N)$.

Due to Nielsen and Chuang, there are at least 03 ways to address the problem that $r' \neq r$ due to $\gcd(k, r) \neq 1$.

- For a randomly chosen $k$ between 0 and $r-1$, it is *quite likely* that $\gcd(k, r) = 1$. This is due to the *prime number theorem* which states that the number of primes less than $r$ is at least $r/2\ln r$. This means that the chance that $s$ is prime, and therefore relatively prime to $N$, is at least $1/2\ln r > 1/2\ln N$. So, by repeating the algorithm $\boxed{2\ln N}$ times we will most likely get $k, r$ that are co-prime, which is what we want.
- Another way to fix this problem is doing more post-processing. If $r' \neq r$ then we know that $r'$ is a factor of $r$ unless $k = 0$. The case where $k = 0$ can be ignore because we can always add a few extra iterations (as discussed in Part (a)). Now, take $a' \equiv a^{r'} \mod N$ and run the algorithm again. The order of $a'$ is then $r/r'$. If we manage to find $r/r'$, then we can compute via $r = r' \times r/r'$. If not, we get some $r''$ which is yet again a factor of $r'$. In this case, we repeat by defining $a'' \equiv a'^{r''} \mod N$ and proceed until termination and compute $r$ at the end. How many iterations does this method require? The answer is at most $\ln r$, which is $\boxed{O(L)}$, since each repetition the order is reduced by a factor of at least 2.
- This third method requires a constant number of trials: we start by repeating order-finding algorithm twice to get two tuples $(k_1', r_1')$ and $(k_2', r_2')$. If $k_1', k_2'$ are co-prime, then $r = \mathrm{lcm}(r_1, r_2)$. How likely is this? The probability that $k_1', k_2'$ are co-prime is given by

$$1 - \sum_q \Pr(q|k_1') \Pr(q|k_2') \geq 1 - \sum_q \Pr(q|k_1) \Pr(q|k_2) \geq 1 - \sum_q \frac{1}{q^2}$$

where the sum is over all primes. The first inequality follows from the fact that if $q$ divides $k_i'$ then it must also divide $k_i$. Moreover, $\Pr(q|k_i) \leq 1/q$. So we get the second inequality. Next, since

$$\int_x^{x+1} \frac{dy}{y^2} \geq \frac{2}{3x^2} \qquad \forall\ x \geq 2,$$

we have that

$$\sum_q \frac{1}{q^2} \leq \frac{3}{2} \int_2^\infty \frac{dy}{y^2} = \frac{3}{4}.$$

So, the probability that $k_1', k_2'$ are co-prime is

$$1 - \sum_q \Pr(q|k_1') \Pr(q|k_2') \geq \frac{1}{4}.$$

This means that the probability of getting the correct $r$ is at least $1/4$.

**3.** For a prime $p$ and two numbers $a, b$ with $1 < a, b < p$, consider the quantum state

$$\frac{1}{\sqrt{p}} \sum_{j=0}^{p-1} |j \mod p\rangle \, |(aj+b) \mod p\rangle \, .$$

The approach to this problem is via the Quantum Fourier Transform (QFT), which we will use to find the period of the map $j \mod p \to (aj+b) \mod p$, which is determined only by $a$. So, the goal is to use the QFT to find $a$. In fact, using only the QFT, we can only find $a$ since the FT is translationally invariant, i.e., the shift due to $b$ will not show up.

(a) Show that if you are given one copy of this quantum state, then with a quantum computer, you can find $a$ with high probability (i.e., with probability going to 1 as $p$ goes to $\infty$).

To start, we apply QFT on the second register and inverse QFT on the first register:

$$|j\rangle \to \frac{1}{\sqrt{Q}} \sum_{k=0}^{Q-1} e^{-2\pi i j k/Q} \, |k\rangle \, .$$

$$|aj+b\rangle \to \frac{1}{\sqrt{Q}} \sum_{\ell=0}^{Q-1} e^{2\pi i (aj+b)\ell/Q} \, |\ell\rangle \, .$$

The output of this operation is

$$\frac{1}{Q\sqrt{p}} \sum_{k=0}^{Q-1} \sum_{\ell=0}^{Q-1} e^{2\pi i [(aj+b)\ell - jk]/Q} \, |k\rangle \, |\ell\rangle \, .$$

Now we measure both registers in the standard basis. The probability of seeing the output $|k\rangle \, |\ell\rangle$ is proportional to

$$\frac{1}{p} \left| \sum_{j=0}^{p-1} e^{2\pi i [(aj+b)\ell - jk]/Q} \right|^2 = \frac{1}{p} \left| \sum_{j=0}^{p-1} e^{2\pi i j [a\ell - k]/Q} \right|^2$$

which is sharply peaked and dominated by the term with $a\ell - k \equiv 0$. In the spirit of quantum phase estimation, we can find what $a$ is given $\ell$ and $k$. This algorithm fails if $k$ or $\ell$ is zero, and it is easy to see that the probability of finding $k\ell = 0$ is $1/p$ and therefore vanishes as $p \to \infty$. So, the probability of finding $a$ goes to 1 as $p \to \infty$.

(b) Show that you can also find $b$ with high probability (i.e., with probability going to 1 as $p$ goes to $\infty$).

To do this problem, we must somehow interchange the roles of $a$ and $b$. To do this, we first need a unitary that sends $j$ to $j^{-1}$. This is not hard since we know $p$ already. The only problem is $j = 0$, but since 0 is its own inverse, we just send 0 to itself. We also have to send $aj + b$ to $a + bj^{-1}$. We can do this by multiplying $aj + b$ by $j^{-1}$ if $j \neq 0$. We cannot do this if $j = 0$.

One way to overcome the $j = 0$ problem is to project out the component $|0\rangle \, |a0 + b\rangle = |0\rangle \, |b\rangle$. This can be done via a subspace measurement. After this process, the input state becomes

$$\frac{1}{\sqrt{p-1}} \sum_{j=1}^{p-1} |j \mod p\rangle \, |aj+b \mod p\rangle \, .$$

The unitary $U_1$ for the first register can be constructed fairly easily once $p$ is known. The unitary for the second register is simply one that does multiplication where the factor $j$ comes from the first register: $U_2 |x\rangle |y\rangle = |x\rangle |xy\rangle$. The application of $U_1$, followed by $U_2$, gives

$$\frac{1}{\sqrt{p-1}} \sum_{j=1}^{p-1} \left| j^{-1} \mod p \right\rangle \left| a + bj^{-1} \mod p \right\rangle = \frac{1}{\sqrt{p-1}} \sum_{j'=1}^{p-1} |j' \mod p\rangle |a + bj' \mod p\rangle .$$

From here, we use the method outlined in Part (a) in order to find $b$. We note that due to the missing $|0a\rangle$ component, the algebra isn't exactly the same. However, the principles apply: after the QFT on the second register and inverse QFT on the first register, we have the quantum state

$$\frac{1}{Q\sqrt{p-1}} \sum_{k=0}^{Q-1} \sum_{\ell=0}^{Q-1} e^{2\pi i[(aj+b)\ell - jk]/Q} |k\rangle |\ell\rangle .$$

Now we measure both registers in the standard basis. The probability of seeing the output $|k\rangle, |\ell\rangle$ is proportional to

$$\frac{1}{p-1} \left| \sum_{j=0}^{p-1} e^{2\pi i[(aj+b)\ell - jk]/Q} \right|^2 = \frac{1}{p-1} \left| \sum_{j=0}^{p-1} e^{2\pi ij[a\ell - k]/Q} \right|^2$$

which peaks whenever $a\ell - l \equiv 0$. Once again in the spirit of quantum phase estimation, we can find that $a$ is given $\ell$ and $k$. For the same reason as Part (a), the probability that the algorithm fails vanishes as $p \to \infty$: the probability that we measure the input state and find it in $|0\rangle |b\rangle$ scales as $1/p$ which vanishes as $p \to \infty$. So, the probability of finding $b$ goes to 1 as $p \to \infty$.

(c) Show that no quantum algorithm can identify $a$ with probability 1.

No quantum algorithm can identify $a$ with probability 1 because some inputs cannot be distinguished from each other with certainty. Consider two pairs $(a_1, b_1) \neq (a_2, b_2)$. Let us compute the overlap between two quantum states associated with these pairs:

$$\begin{aligned} \frac{1}{p} \left( \sum_{j=0}^{p-1} \langle j| \langle a_1 j + b_1| \right) \left( \sum_{k=0}^{p-1} |k\rangle |a_2 k + b_2\rangle \right) &= \frac{1}{p} \sum_{j=0}^{p-1} \sum_{k=0}^{p-1} \langle j|k\rangle \langle a_1 j + b_1 | a_2 k + b_2\rangle \\ &= \frac{1}{p} \sum_{j=0}^{p-1} \langle a_1 j + b_1 | a_2 j + b_2\rangle . \end{aligned}$$

Now each summand is either 0 or 1. We claim that there always exists some $j$ for which $a_1 j + b_1 \equiv a_2 j + b_2$. To see this, notice that since $p$ is prime, $a_1 - a_2$ must be relatively prime to $p$, and therefore $j = (a_1 - a_2)^{-1}(b_2 - b1)$. And we're done. Note that I'm using the $(a_1 - a_2)$ and $(b_1 - b_2)$ notations quite loosely. All of the operations above must be in modular arithmetic of course.

For this reason, the input states spanned by values of $a$ and $b$ are not necessarily mutually orthogonal. No quantum algorithm can therefore distinguish them (i.e. find $a$) with probability 1 without breaking unitarity.