Name: **Huan Q. Bui**
Course: **8.370 - QC**
Problem set: **#7**
Due: Wednesday, Nov 9, 2022
Collaborators/References:

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

$$U\,|u \mod 21\rangle = |2y \mod 21\rangle$$

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 $e^{2\pi ia/b}$. 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

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

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

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

(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.

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 $r' = r$?

The answer is that 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 and how likely it occurs?

(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.)

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 $r = \text{lcm}(r_1, r_2)$. 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

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

(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 ∞).

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

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