Problem Set 2

1. Generalized Bernstein-Vazirani Let M, n be a positive integers. Let $s \in \mathbb{Z}_M^n$ and define the function $f_s : \mathbb{Z}_M^n \to \mathbb{Z}_M$ by $f_s(x) = \langle s, x \rangle \mod M$. Given access to an oracle O_{f_s} which for $x \in \mathbb{Z}_M^n, b \in \mathbb{Z}_M$ acts as $O_{f_s}|x\rangle|b\rangle = |x\rangle|b + f_s(x) \bmod M\rangle$, design a quantum algorithm that computes s with one application of O_{f_s}.

Hint: You may want to generalize the "phase-kickback trick" to show with the oracle O_{f_s} you can also implement an oracle O'_{f_s} with the behavior

$$O'_{f_s}|x\rangle|b\rangle = \omega^{-f_s(x)\cdot b}|x\rangle|b\rangle$$

where $\omega = e^{2\pi \mathrm{i}/M}$.

Bonus: What kind of errors in the oracle can your algorithm tolerate (analogous to what we saw in problem 7 of problem set 1)?

2. Continued fractions In the classical post-processing of Shor's period finding algorithm we have a fraction b/N and want to find the best rational approximation to this number whose denominator is at most M. In lecture we said this can be done in polynomial time as the task can be written as a two-variable integer linear program. Now we see a direct way to do this via continued fraction expansion. A nice discussion of continued fractions, including all the material below, can be found in Chapter 10 of Hardy and Wright's An introduction to the theory of numbers.

A finite continued fraction is an expression of the form

$$a_0 + \cfrac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \frac{1}{\cdots + \frac{1}{a_t}}}}} \ .$$

We will denote this number by $[a_0, \ldots, a_t]$. For $0 \le j \le t$ we call $[a_0, \ldots, a_j]$ the j^{th} convergent to $[a_0, \ldots, a_p]$. A continued fraction $[a_0, \ldots, a_t]$ is called *simple* if $a_1, \ldots, a_p$ are all positive integers (a_0 can be non-positive). Every rational number can be represented by a finite simple continued fraction.

Here is an algorithm to find such a representation. Let x be a positive rational number. Then set

$$
\begin{aligned}
a_0 = \lfloor x \rfloor, &\quad x_1 = \frac{1}{x - a_0} \\
a_1 = \lfloor x_1 \rfloor, &\quad x_2 = \frac{1}{x - a_1} \\
a_2 = \lfloor x_2 \rfloor, &\quad x_3 = \frac{1}{x - a_2} \\
&\cdots
\end{aligned}
$$

The essential principle at work here is that $x = a_0 + \frac{1}{a_1'}$ where $a_1' = \frac{1}{x - a_0}$. Then since $[a_0, [a_1, \ldots, a_t]] = [a_0, a_1, \ldots, a_t]$ our task becomes to find a continued fraction expansion of a_1' which we do by the same procedure.

One can also find an inductive expression for $[a_0, \ldots, a_j]$. If

$$
\begin{aligned}
p_0 &= a_0, & p_1 &= a_1 a_0 + 1, & p_j &= a_j p_{j-1} + p_{j-2} \\
q_0 &= 1, & q_1 &= a_1, & q_j &= a_j q_{j-1} + q_{j-2}
\end{aligned}
$$

then $[a_0, \ldots, a_j] = \frac{p_j}{q_j}$ and this is in lowest terms. Note that $q_j \geq 2q_{j-2}$ thus q_j increases at least exponentially. An important property of the continued fraction expansion for the application in Shor's algorithm is that if

$$
|x - \frac{c}{d}| \leq |x - \frac{p_j}{q_j}|
$$

then $d \geq q_j$.

Now the questions:

1. Find the continued fraction expansion of $\frac{527}{1024}$.

2. Look at the j^{th} convergents of your expression and make a conjecture about the even and odd numbered convergents (you do not need to prove it).

3. (Optional but could be helpful for Problem 3) Write a program in any language to compute a continued fraction of an input number up to a given accuracy.

3. Factoring 21 Let's factor the number 21 using Shor's algorithm.

1. List all numbers in $\mathbb{Z}_{21}$ that are relatively prime to 21. These are the elements of the multiplicative group $\mathbb{Z}_{21}^\times$. Compute the order $\mathrm{ord}_{21}(x)$ of all elements in $\mathbb{Z}_{21}^\times$.

2. Choose an $x \in \mathbb{Z}_{21}^\times$ with $\mathrm{ord}_{21}(x) = 6$. See that $\gcd(x \pm 1, n)$ gives a nontrivial factor of n.

3. Now let's simulate finding the order of $f(j) = x^j \bmod 21$ for the x you chose in the last step. Using the Octave FTperiod program [1] `https://github.com/troyjlee/qalgo/`

[1] Currently I have only added the sampling functionality to the Matlab/Octave program. If I have time I will also add it to the python version. Octave programs can be run online at `https://octave-online.net/`.

`tree/main/CODE` with $N = 21^2, s = 6$. This simulates randomly sampling a state $|g_t\rangle$ and measuring $F_N|g_t\rangle$ to see an index b. Use continued fraction expansion on b/N and see if you can recover $\mathrm{ord}_{21}(x)$. It may take several attempts. Record the values you see and how many attempts it takes.

4. Assumptions Where in the proof of correctness of Shor's algorithm for the general period finding problem with a function $f : \mathbb{Z}_N \to [M]$ do we use the assumption that $N > M^2/2$? What can go wrong without this assumption?

5. Cosets Let G be a finite group and $K, L \leq G$ subgroups of G. For $a, b \in G$ let $aK = \{a \cdot k : k \in K\}$ be a left coset of K and bL similarly be a left coset of L. If $d = |K \cap L|$ show that $|aK \cap bL| \in \{0, d\}$.

6. Finding all ones Let $N = 2^n$ and $x \in \{0,1\}^N$ and *assume you know* that x has k many ones.

1. In lecture we showed how to find an $i \in N$ such that $x_i = 1$ with constant probability by a quantum algorithm after $O(\sqrt{N/k})$ many queries to x. Show how to boost this success probability to $1 - 1/N^2$ using $O(\sqrt{N/k}\log(N))$ many queries to x.

2. Give a quantum algorithm to find *all* the ones in x with constant probability after $O(\sqrt{kN}\log(N))$ many queries to x.

7. Exact searching Do Exercise 4 in Chapter 7 of Ronald de Wolf's lecture notes `https://arxiv.org/abs/1907.09415`. For part (c) you may assume you have access to the phase oracle $O_{f,\pm}$ for f and may use extra ancillas and any elementary gates you like.