Header styling inspired by CS 70: https://www.eecs70.org/

1 Quantum Order Finding/Shor's Algorithm

1.1 From Nielsen-Chuang

- Basically quantum phase estimation except with the matrix $U|y\rangle = |xy \pmod N\rangle$. By convention, we say that U only acts when $0 \leq y \leq N-1$, and when $N \leq y \leq 2^N - 1$, that $U|y\rangle = |y\rangle$.
- The eigenstates of U are given by

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(\frac{-2\pi i s k}{r}\right) |x^k \pmod N\rangle$$

  when $s \in \{0, \ldots, r-1\}$. This is because:

$$U|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left(\frac{-2\pi i s k}{r}\right) |x^{k+1} \pmod N\rangle = \exp\left(\frac{2\pi i s}{r}\right) |u_s\rangle$$

  Why is the eigenvalue not $\exp\left(-\frac{2\pi i s}{r}\right)$? Shouldn't we be multiplying by the negative to get the $k+1$?

  It might be becuase we're re-indexing by going *down*, not *up* (i.e. we're going from k to $k-1$ and not k to $k+1$. Therefore, we multiply by a positive factor.

- We can perform modular exponentiation to perform the controlled-U^{2^j} operation, and we can use the fact that

$$\frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = 1$$

  in order to prepare the state $|u_s\rangle$, without knowing r at all.

1.2 From Lecture Notes

- Use two registers, the first of which has K qubits such that $Q = 2^K$, and $N^2 \leq Q \leq 2N^2$ (so we have Q basis states). The second register has at least $n = \log_2 N$ qubits, with N basis states.
- We first initialize both registers in the state $|0\rangle \otimes |0\rangle$

  So do we mean that register 1 has $|0\rangle^{\otimes K}$?

- Source register is transformed to an equal superposition over all Q basis states

$$H^{\otimes K}|0\rangle = \frac{1}{\sqrt{2^K}} \sum_y |y\rangle$$

  You can also accomplish the same thing by applying the Quantum Fourier transform:

$$|0\rangle \to \frac{1}{Q} \sum_{q;=0}^{Q-1} |q'\rangle$$

  Either way, we end up with the final state:

$$\frac{1}{\sqrt{Q}} \sum_{q=0}^{Q-1} |q\rangle \otimes |0\rangle$$

- Apply a gate U_a that implements the Quantum modular exponentiation, as in $q \to f(q) = a^q \pmod N$. We choose a randomly. Note that we don't really care about the implementation fo this very much. This function $f(q)$ has r as a period, and we want to find r. Applying f to the values of the first register and storing the values in the second gives us:

$$\frac{1}{\sqrt{Q}} \sum_{q=0}^{Q-1} |q\rangle \ |a^q \mod N\rangle$$

  All Q values of the function $f(q)$ are computed in parallel, so the value of r will certainly appear somewhere.

- Measure the second register, and we get some value, say $f(q_0)$. If $Q = mr$, then there are m different values of q that give us the result $f(q)$, so those are the states that remain. So, the state is now:

$$\frac{1}{\sqrt{Q/r}} \sum_{j=0}^{Q/r-1} |jr + q_0\rangle \ |f(q_0)\rangle$$

  Recall that we took the values from register 1 and stored $f(q)$ in register 2, so therefore the second register has fully collapsed, but the first register still contains multiple possible values, specifically multiples of the order plus q_0.

- Apply the Fourier transform modulo Q on the first register. This gives us:

$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \omega^{kq_0} \ |k\frac{Q}{r}\rangle$$

  So therefore, the total state is:

$$\frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \omega^{kq_0} \ |k\frac{Q}{r}\rangle \ |f(q_0)\rangle$$

- Measure register 1, which gives us a value $C = k\frac{Q}{r}$, where k is some number between 0 and $r-1$ (since we only have access to one of these values). Repeating this procedure will allow us to get all the values of k, and eventually we can get r.

- When we can't assume that $Q = mr$, then we get a modified analysis that uses continued fractions in order to determine r.

- Once we have the order, we can actually just factor and some clever number theory results to factor our number N.

2 Grover's Algorithm

2.1 Nielsen-Chuang

- Problem framed as a "promise" problem: where we're given a function f that takes in an integer x, and $f(x) = 1$ only when x is a solution to the search problem, and $f(x) = 0$ otherwise.

- First, suppose f is implemented in terms of an "oracle", which can recognize solutions to the search problem. We can define the oracle O in temrs of:

$$|x\rangle \ |q\rangle \xrightarrow{O} |x\rangle \ |q \oplus f(x)\rangle$$

  Where $\oplus$ denotes addition modulo 2. This means that if $f(x) = 1$, the qubit $|q\rangle$ is flipped, and otherwise it's untouched. Therefore, to check whether $|x\rangle$ is a solution, we send in $|x\rangle \ |0\rangle$, and see if the output is $|x\rangle \ |1\rangle$; if yes, then we know that x is the solution.

- Just like the Deutsch-Jozsa algorithm, we will apply the oracle to the state

$$|x\rangle \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

  If x is not a solution to the problem, then it will leave the state unchanged. If x is the solution, then it will flip the 0 to a 1 and vice versa, so we'd get the state:

$$|x\rangle \left(-\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)\right)$$

- Since it flips by a negative sign when $f(x) = 1$, then we can write this as:

$$|x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right) \xrightarrow{O} (-1)^{f(x)} |x\rangle \left(\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right)$$

- With this in mind, we can actually get rid of the oracle bit entirely:

$$|x\rangle \xrightarrow{O} (-1)^{f(x)} |x\rangle$$

  With M solutions, then we need only $O(\sqrt{N/M})$ times in order to obtain a solution.

- To do the search itself, we first start with the state

$$|\psi\rangle = \frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle$$

  We then use a Grover operator called G repeatedly, which works like this:

  - Apply the oracle O
  - Apply the Hadamard transform $H^{\otimes n}$
  - Perform a conditional phase shift:

$$|x\rangle \to -(-1)^{\delta_{x0}} |x\rangle$$

    What is δ_{xo}?

    Apparently, the phase shift basically is $|0\rangle \to |0\rangle$, and $|x\rangle \to -|x\rangle$ for all $x > 0$.

  - Apply the Hadamard transform $H^{\otimes n}$

- For simplicity, we can write $G = 2(|\psi\rangle \langle\psi| - I)O$. Geometrically, we can interpret Grover's algorithm as a rotation in the 2-dimensional space spanned by the starting vector $|\psi\rangle$, and the vector that is the superposition of the solutions.

- In essence, $G|\psi\rangle$ rotates the vector $|\psi\rangle$ closer to $|\beta\rangle$, which is a superposition of the solutions. Therefore, once we get close to $|\beta\rangle$, making a measurement gives a solution to the search problem with high probability.

2.2 Lecture Notes

- Suppose we're given a function $f : \{0, \ldots, N-1\} \to \{0, 1\}$, with the *promise* that there is some x_0 such that $f(x_0) = 1$.

- Quantumly, f is implemented as an "oracle" Q_f, with the properties:

$$Q_f |x\rangle = \begin{cases} -|x\rangle & f(x) = 1 \\ |x\rangle & f(x) = 0 \end{cases}$$

  So in other words, we can write $Q_f(x) = (-1)^{f(x)} |x\rangle$.

- Before Grover, we assume an algorithm of the following form:

$$U_0, Q_f, U_1, Q_f, U_2, Q_f, \ldots, U_T$$

  But we don't know which entry will be marked by f, so it's best for us to start with a maximally symmetric state:

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

  This is the same as $H^{\otimes n} |0\rangle^{\otimes n}$ if $N = 2^n$. But, the probability of measuring the right $|x_0\rangle$ in this case is $\frac{1}{N}$.

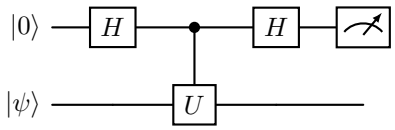
- Grover's trick is to use $U = H(2|0^n\rangle \langle 0^n| - I_n)H$. If we repeat this $\frac{\pi}{4}\sqrt{N}$ times, then we get a probability of nearly 1 of measuring $|x_0\rangle$.

- The idea is that we know that by creating a maximally superimposed state, that when the oracle acts on the state it will change one of the values. Then, we can use Grover's operator $2|u\rangle \langle u| - I$ in order to amplify that state, since it gets us closer to the state of solutions that we want.

3 Phase Estimation

3.1 Lecture Notes

- Suppose we had a matrix U that transforms as: $U|\psi\rangle = e^{2\pi i\theta}|\psi\rangle$. In other words, it adds some global phase to the state.
- We basically need to come up with a way to figure out what this θ value is. To do this, we have some state $|\psi\rangle$, and another state that starts out in $|0\rangle$:



  After the first two gates, we have:

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\phi}|1\rangle) \otimes |\psi\rangle$$

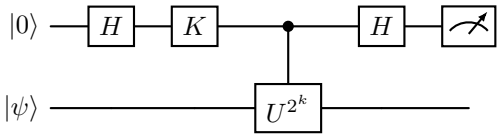
  why do we even care about $|\psi\rangle$, if we're not going to use it?

- Then looking at the first qubit entirely, if we perform another Hadamard transform (or equivalently, measure in the Hadamard basis), then it's not hard to show that the probabilities of getting 0 or 1 are:

$$P(0) = \frac{1+\cos(2\pi\phi)}{2}, \quad P(1) = \frac{1-\cos(2\pi\phi)}{2}$$

  The issue with something liek this, is that the distributions aren't as great (ϕ vs $1-\phi$) , so we want to modify our circuit a little bit.

- The modified circuit looks like this:



  Basically, the only thing we changed was that we added in this new gate K.

- There is a way of doing this with Fourier transforms, but I'm not sure if I have enough time to understand it.

How does this way of phase estimation coincide with the one we did in homework, where we measured in the Hadamard basis, and applied $K = S$?

Measuring in the Hadamard basis is equivalent to performing a Hadamard transofrm (i.e. Hadamard gate), then measuring in the computational basis. The probabilities are the same.