An Introduction to Quantum Computing: Notes

Christian Zhou-Zheng

July 28, 2023

1 Day 1

Begin with model of the atom - planetary model, orbitals represent likelihood that the electron will be in some area/at some energy level. Heisenberg uncertainty principle and such; basic atomic chemistry.

Quantum: idea of quantization/discretization (discrete intervals rather than "sliding scale") of certain properties on very small scales; you should vaguely remember this from chem. Quantum effects have few macroscopic effects, but are very important on the atomic scale.

Brought up Zeno's paradox, involving infinite sums of infinitely small quantities. Idea of quantum mechanics is that at some point you can't get smaller - there's a minimum (Planck units), so Zeno's paradox is incompatible with actual mechanics.

Quick fling out to computer science: we made a hierarchy of basic computer components, from applications to the OS to CPU/memory to logic gates to transistors. Transistors are where the 1s and 0s come from - they either have an electrical current or not, 1 or 0. We talked about alternatives to each of the five mentioned components.

Tying in everything is probability and amplitude. Amplitude is the likelihood of a certain event occurring; amplitudes can be negative! The *probability* is the square of the absolute value of the amplitude, and the sum of all possible probabilities must equal 1: $\sum_{i=0}^{N} p_i = 1$. This leads to weird looking math: for example, E_1 having amplitude $\frac{1}{\sqrt{2}}$ and E_2 having amplitude $-\frac{1}{\sqrt{2}}$ will have a sum of probabilities equal to 1!

Quick notation note: states (e.g. not raining vs. raining) are often notated using brackets, and wave functions/quantum states are denoted with psi: $|\Psi\rangle = \alpha |E_1\rangle + \beta |E_2\rangle$ for two events/states $|E_1\rangle$, $|E_2\rangle$ where α, β are the respective amplitudes. We use the uncertainty principle here to extend this to the electron: the state of the electron $|\Psi\rangle$ comprises its position and velocity (our $|E_1\rangle$, $|E_2\rangle$), and we can only measure one!

NOTE: the probability of a state is the square of the **absolute value** of the amplitude, because the amplitude is often a complex number!!!

The fact that probabilities sum to 1 is important - graphing the different states models the unit circle, sphere, or whatever in higher dimensions. This conceptually links this idea to wave functions (*amplitude?*) - $\sin^2(x) + \cos^2(x) = 1$ models a circle as well!

This goes back to actual quantum *computing* because we can have the two states being 0 and 1: we don't know which one it is, but we can measure it (in its superposition) and it collapses to give either a 1 or a 0! The challenge here is how to define/create algorithms that manipulate these states such that these states will do the intended calculations efficiently and accurately.

2 Day 2

Quick recap of discreteness and sets to start class - continuous vs discrete.

Today is WAVES! Yesterday we did a bit on wave functions - probability and amplitude of quantum functions. But they're not waves.

Key terms: amplitude, interference, superposition. These are all drawn from the language of waves - a wave alone has wavelength and amplitude, while we draw the latter two from interference. Looking at two waves colliding head-on, they share the position and don't bounce off or anything - they pass right through. The fact that they have no issue with being in the same place is called superposition. You know what interference is at this point - constructive and destructive, etc etc.

We go over light a little - history of particle to wave to particle, culminating in the idea of wave-particle duality. Recall yesterday's idea of collapsing: sending quantum particles through a double slit experiment, the particles act quantum-ly and only collapse when hitting the wall, where they form an interference pattern as if they were waves. However, if you set up an observer by the slits, it collapses the particles early - but they return to acting quantum-ly after you stop observing them, through the slit, and it forms a normal distribution you'd expect if they were particles. Again, this idea of collapsing the superposition comes up.

We use wave-like properties (wave functions; psi) in probabilities to explain this stuff. $\psi(x, t)$ is the probability amplitude α of finding the object in question at position x at time t. In the observed double slit experiment, the probability of finding the photon at slit 1 $P_1 = |\psi_1|^2$, and same for slit 2. It then starts a "new" wave function.

In the non-observed experiment, we have one wave function for the whole thing: $P_{12} = |\psi_1 + \psi_2|^2 = |\psi_1^2 + \psi_2^2 + 2\psi_1\psi_2|$ (assuming both slits are equally likely), based on the superposition of the two possibilities - assuming both are happening at the same time! In the latter form, $2\psi_1\psi_2$ is the interference term - it accounts for the interaction between the two wave functions. The squared terms are kind of like the photon going through each slit, whereas the interference term is like the photon interfering with itself - you've seen it before, the quantum object breaks into multiple wave functions, interferes with itself, and collapses!

Wave functions are really probability waves - they're not physical. Measuring them collapses them into one state, and we lose all information about the other amplitudes!

We finish the main portion of the class with a video on Schrodinger's cat and segue into a bit on

basic complex numbers - all elementary.

3 Day 3

Pointlessly, you can treat quantum objects like normal transistors by using specific manipulations.

The most common quantum object used as a transistor replacement: superconducting circuits, hence the incredibly low temperatures required.

Today's lecture is on bits and qubits, the basic units of information. You know what bits are - binary digits, 0 or 1, stored in transistors. We go over binary representations a bit, basic compsci things, as well as conversion between bases. (Fun fact: octal is used for file permissions in Unix-like OSs.)

Talked a bit about character encodings for some reason: ASCII, UTF-8, Unicode. My initials in ASCII are apparently 1000011 1011010 10111010.

Logic gates! How to manipulate bits. You know these (from Redstone, lmao) - AND NOT OR NAND NOR XOR XNOR etc etc. We start drawing diagrams and such. There are four basic *operations* you can do on a single binary input: flip it, do nothing (identity transformation), or force it into a 1 or 0. We look at truth tables, drawing out logic gates, etc etc.

Unfortunately, it's difficult to draw out the figures of different gates on LaTeX. I'll just describe them: AND is a D shape, NOT is a triangle with a circle on the point, OR is a D shape with a concave flat side, XOR is OR with another line on the concave side, and the N-etc. gates are the equivalent gate with a circle on the round side. I'm sure I don't have to draw out the truth tables for the gates - you know these by heart.

4 Day 4

Finally introduced to qubits! We can use things like electron properties or superconductors as quantum objects we use as qubits. Talked a bit about light, properties of photons, quick example of polarization through sunglasses - electric fields in wave form around a photon, etc, to show photon polarization is a quantum object too. Quickly went over how the axes representing 1 and 0 have to be perpendicular, since they have to be absolutely exclusive.

We played around with polarized sheets of paper, like sunglass lenses - when placed perpendicularly, they block all light, but if you put a diagonal one between the two you can still see some! I posited a theory of vectors and such, but it's a matter of quantum mechanics - polarizing sheets work like measurements, collapsing photon states.

By measuring in a particular direction, we ask the photon what "direction" it's going, and that forces it to collapse into a state from its former superposition. It has quantum states like any other: $|\Psi_P\rangle = \alpha|V\rangle + \beta|H\rangle$, where V and H are vertical and horizontal polarization, respectively.

It's both horizontal and vertical, just to different amplitudes, and when measured it can **only** be

one or the other when using these two **bases** (singular, *basis*) - even if it's diagonal (kind of both), it collapses into one or the other with amplitude α or β, respectively.

Looking at a scenario with a first vertical polarizing sheet, the photon begins with state $|\Psi_0\rangle$, unknown, and passes through the vertical polarizing sheet (checking for state $|V\rangle$) with probability $|\alpha|^2$. In this case, it is in state $|\Psi_1\rangle = 1\,|V\rangle + 0\,|H\rangle$ (technically, since our requirement is that $|\alpha|^2 + |\beta|^2 = 1$, α could be any value on the complex unit circle - but that's irrelevant for reasons discussed shortly).

So, if we try to make it pass through another vertical sheet, it should pass through with probability $|\alpha|^2 = |1|^2 = 1$, and if we try to pass it through a horizontal sheet, it should pass through with probability $|\beta|^2 = |0|^2 = 0$, as expected! Note that the state $|\Psi_2\rangle$ after a vertical sheet is $|V\rangle$ and after a horizontal sheet 0.

However, if we pass it through a diagonal sheet, our equation no longer helps us. We must change the bases to diagonal and anti-diagonal, which are perpendicular to each other and still satisfy exclusivity. $|\Psi_P\rangle = \alpha\,|\nearrow\rangle + \beta\,|\nwarrow\rangle$. Because of this, we need to obtain the proper α and β values for the new basis system, done by treating them like a vector and rotating the axes of the plane - which is done by matrix multiplication, since matrices can be thought of transformations/functions on vectors!

Anyway, assuming our new bases are at $\frac{\pi}{4}$ radians from the original, $|V\rangle$ in our original basis system (which had $\alpha = 1, \beta = 0$) now has $\alpha = \frac{\sqrt{2}}{2}, \beta = \frac{\sqrt{2}}{2}$. This means that the probability of passing through a diagonal up/right sheet is $|\alpha|^2 = \frac{1}{2}$ - but $|\Psi_2\rangle$, the *state* of the photon after the second sheet, is rather $|\nearrow\rangle$!

Now, we pass the photon in state $|\Psi_2\rangle$ (assuming it's gotten this far - which it has, with probability $\frac{|\alpha|^2}{2}$ from $|\Psi_0\rangle$) through a horizontal filter, checking for state $|H\rangle$. Again, we rewrite $|\Psi_2\rangle$ in terms of the new basis system, and find that $\alpha = \frac{\sqrt{2}}{2}, \beta = \frac{\sqrt{2}}{2}$, so the probability of passing through a horizontal sheet is $|\beta|^2 = \frac{1}{2}$ - and $|\Psi_3\rangle = |H\rangle$! Mathematically, the photon had a probability of $\frac{|\alpha|^2}{4}$ of getting to this state from $|\Psi_0\rangle$, with α being the value from $|\Psi_0\rangle$ - this is how much light passes through this seemingly impossible 3-filter system!

The basis system used by the quantum phenomenon depends on the axis of the measurement! The two bases are that axis and its perpendicular axis. This is kind of hard to wrap your head around. Also, this problem works in 2 dimensions, nice and perpendicular, because the electric field has to be perpendicular to the direction of light propagation, which happens in the third dimension.

Quick note - photons are the quanta of light, indivisible units. They can't split up or anything, so they follow quantum mechanics to get the effects we observe.

We end class with a relaxed half hour about linear algebra - finally getting into new math topics (to me)! It's all review, so reference the QSYS math packet for this.

But do brush up on vector spaces - they're an abstract idea somewhat difficult to grasp. It's a set + operations (a group), such that the set is closed under vector addition and scalar multiplication -

that is, adding two members of the set produces another member of the set, and same for multiplying by any scalar. Two other requirements are the existence of the zero vector $\vec{0}$ (which contains entirely zeroes, as the name suggests) and the existence of an additive inverse for every vector $\vec{v}$ in the set $-\vec{v}$ such that adding them together yields the zero vector.

Matrices are also a new idea, similar to 2D arrays in computer science. Members of a m by n matrix M are denoted with subscripts: row, then column, like 2D array indices. For instance, a 2x3 matrix: $\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \end{bmatrix}$. An easy way to remember the rows come first is thinking like a building: first you take the elevator to the floor you want, then walk down the hall.

Adding two matrices $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and $N = \begin{bmatrix} e & f \\ g & h \end{bmatrix}$ is done by adding the corresponding elements: $M + N = \begin{bmatrix} a+e & b+f \\ c+g & d+h \end{bmatrix}$. Scalar multiplication is done by multiplying each element by the scalar: $cM = \begin{bmatrix} ca & cb \\ cc & cd \end{bmatrix}$.

Multiplying two matrices is NOT commutative, unlike the operations mentioned earlier (but is associative and distributive). It requires the number of columns in the first matrix to be equal to the number of rows in the second matrix, and the resulting matrix will have the same number of rows as the first and the same number of columns as the second: strictly, multiplying a m by n matrix with a n by p matrix results in a m by p matrix. The elements of the resulting matrix are found by multiplying the elements of each row of the first matrix by each element of the column of the second matrix, and adding them together: for instance, with the same M and N, $MN = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}$.

Thinking of matrices as operators on vectors, we can see that multiplying a matrix by a vector results in a vector. For instance, $M \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax+by \\ cx+dy \end{bmatrix}$. This is a linear transformation, and is the basis (no pun intended) of linear algebra. It's also worth pointing out that matrices are a transformation between dimensions of vectors: the original dimension is the number of columns in the matrix, and the resulting dimension is the number of rows in the matrix.

A quick note about compositions of functions: for $f(\vec{v}) = M\vec{v}, g(\vec{v}) = N\vec{v}, f(g(\vec{v})) = M(N\vec{v}) = MN\vec{v}$.

We also define linear functions (hence *linear* algebra): $f(x+y) = f(x) + f(y), cf(x) = f(cx)$. The same goes for functions (matrices!) over vectors $\vec{v}$, so matrices are linear functions.

Then there are things like the matrix conjugate and transpose: taking the conjugate of a matrix $\overline{M}$ is just taking the conjugate of each element in the matrix, and the transpose M^T is swapping the rows and columns. The conjugate transpose is the conjugate of the transpose, and is denoted with a dagger: $M^\dagger = \overline{M^T}$. Watch out for notation errors - be precise!

5 Day 5

Opening question: how do you even store information in a photon or other quantum object? Classically, it's either current ON or OFF.

We begin with a reminder about wave functions and the fact that amplitudes, where information is stored, are complex numbers: $\alpha = a + bi, \beta = c + di$. Since $|\alpha|^2 + |\beta|^2 = 1$, we can rewrite this as $a^2 + b^2 + c^2 + d^2 = 1$. This is a hypersphere in 4D space, and the surface of the sphere is the set of all possible states of the photon.

We then go over normalizing vectors and the imaginary unit circle, which ties into polar form (which we learned some time ago in the readings): distance + angle. For our unit circle vectors, the distance is always 1.

Aha, and we're finally to more abstract computations! Rather than horizontal and vertical, we now use two arbitrary exclusive bases: 0 and 1, for example: $|\Psi_P\rangle = \alpha|0\rangle + \beta|1\rangle$. However, we can still only manipulate α and β.

This notation $|K\rangle$ is called a Dirac ket, and it's notation for a basis of a system we're trying to measure. Now we'll start treating Dirac kets as vectors, and we operate on them using matrices. For instance, $|0\rangle = \begin{bmatrix}1\\0\end{bmatrix}$ and $|1\rangle = \begin{bmatrix}0\\1\end{bmatrix}$: the upper number is α and the lower number is β in a corresponding state. For a generic state $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, we can write $|\Psi\rangle = \alpha \begin{bmatrix}1\\0\end{bmatrix} + \beta \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}\alpha\\\beta\end{bmatrix}$. Talking compsci, it's like zero-indexing: the zero ket has element 0 equal to 1, the one ket has element 1 equal to 1.

Now for a few conceptual questions. First, can you prove $|0\rangle$ and $|1\rangle$ are perpendicular? Yes, they're orthogonal - their dot product is 0. $|0\rangle \cdot |1\rangle = \begin{bmatrix}1\\0\end{bmatrix} \cdot \begin{bmatrix}0\\1\end{bmatrix} = 0$. They're *linearly independent* - you can't write one as a linear combination of the other. Therefore, they're a valid basis system for the complex plane. Note that any n-dimensional vector can be written as a linear combination of n basis vectors of that dimension.

Now can you prove that any state $|\Psi\rangle = \begin{bmatrix}\gamma\\\delta\end{bmatrix}$ can be written as a linear combination of the two basis states $|0\rangle, |1\rangle$? Yes, it's just $\gamma|0\rangle + \delta|1\rangle = \gamma \begin{bmatrix}1\\0\end{bmatrix} + \delta \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}\gamma\\0\end{bmatrix} + \begin{bmatrix}0\\\delta\end{bmatrix} = \begin{bmatrix}\gamma\\\delta\end{bmatrix}$. (Formally, this is backwards - and it assumes the original wave function was already normalized.)

Is the state $|\Psi\rangle = (\frac{1}{2} + \frac{1}{2}i)|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$ normalized? Looking at a, b, c, and d, we see: $\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2 + \left(\frac{1}{\sqrt{2}}\right)^2 + (0)^2 = 1$, so yes. This is again because $|\alpha|^2 + |\beta|^2 = 1$, and the square of the modulus of a complex number is the sum of the squares of its real and imaginary parts.

We also define the two intermediate bases: $|+\rangle$ and $|-\rangle$, defined as $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. We can then write $|0\rangle = \frac{1}{\sqrt{2}}(|+\rangle + |-\rangle)$ and $|1\rangle = \frac{1}{\sqrt{2}}(|+\rangle - |-\rangle)$ - conceptually,

you can see $|0\rangle$ is an equal superposition of $|+\rangle$ and $|-\rangle$, and $|1\rangle$ is an equal superposition of $|+\rangle$ and $-|-\rangle$. (Apply these to $+\rangle$ and $|-\rangle$!)

Remember, this is braket notation - like there are Dirac kets $|K\rangle$, there are Dirac bras $\langle B|$. The bra is the conjugate transpose of the ket, so $\langle K| = |K\rangle^\dagger$. Combine the two and you get a braket $\langle B|K\rangle$ which represents the inner/dot product of $|B\rangle$ and $|K\rangle$: $\begin{bmatrix}\overline{B_1} & \overline{B_2}\end{bmatrix} \cdot \begin{bmatrix}K_1 \\ K_2\end{bmatrix} = \overline{B_1}K_1 + \overline{B_2}K_2$, a scalar quantity (remember that the inner/dot product requires the conjugate of the vector, something that isn't brought up in precalc/physics class - since you never use complex numbers!).

For example, $\langle +|-\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\end{bmatrix} \cdot \frac{1}{\sqrt{2}}\begin{bmatrix}1 \\ -1\end{bmatrix} = \frac{1}{2}\begin{bmatrix}1 & 1\end{bmatrix} \cdot \begin{bmatrix}1 \\ -1\end{bmatrix} = \frac{1}{2}(1-1) = 0$ as expected - they're orthogonal. If the inner product of a vector with itself is 1, on the other hand, it's normalized. Furthermore, if the inner product of two vectors/kets K_i and K_j is 1, they're equal.

There's also the outer product, $|K\rangle\langle B|$, which equals $\begin{bmatrix}K_1 \\ K_2\end{bmatrix} \cdot \begin{bmatrix}\overline{B_1} & \overline{B_2}\end{bmatrix} = \begin{bmatrix}K_1\overline{B_1} & K_1\overline{B_2} \\ K_2\overline{B_1} & K_2\overline{B_2}\end{bmatrix}$.

Notice that the sum of the outer products with themselves of $|0\rangle$ and $|1\rangle$ - that is, $|0\rangle\langle 0| + |1\rangle\langle 1|$ - equals the identity matrix in two dimensions: $\begin{bmatrix}1 & 0 \\ 0 & 0\end{bmatrix} + \begin{bmatrix}0 & 0 \\ 0 & 1\end{bmatrix} = \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix}$. This actually holds for any two basis vectors, but keep in mind that the identity matrix *depends on your choice of basis*!

The outer product of any state with itself is known as the *projection* operator, called such because it's a matrix and is therefore an operator on another vector. For example, $|U\rangle\langle U|$ is the projection operator for $|U\rangle$; it projects any vector onto the $|U\rangle$ axis. It doesn't *rotate* the vector, just takes the component of the target vector that's along the $|U\rangle$ axis. This is easiest to see with $|0\rangle$ and $|1\rangle$ - the projection operators for each just keep the x- and y- components of the vector, respectively, and discard the rest. You can also think of this as calculating the scalar quantity of how much $|U\rangle$ is in the target vector, and then multiplying it by the axis vector $|U\rangle$ to get the projection.

6 Day 6

We just went to the Brown Design Workshop and played around with ICs and classical logic gates - building other gates out of NANDs.

7 Day 7

Argh... linear algebra review. I think I'm fine enough on it, but that's what we're doing today.

REMEMBER ROW-COLUMN ORDER. When we say a matrix is "m by n," we mean it has m rows and n columns. During matrix multiplication, the first one must have the same number of columns as the second has rows. Thinking of it like a wood chipper is helpful: each column of the second "falls on" each row of the first and creates a new element in the product through summing.

It's also worth noting (I don't think I mentioned this before) that, in serving as operations on

vectors, matrices can transform vectors from one dimension to another: a m by n matrix operating on a n-dimensional vector outputs a m-dimensional vector. However, it's easy to see that matrices aren't always surjective (although they're injective).

Matrices are, again, associative and distributive but not commutative.

Review of "linearity:" $f(x+y) = f(x) + f(y), cf(x) = f(cx)$. Matrices follow these rules.

Apparently "dot product" is mostly used for real vectors and "inner product" is mostly used for complex vectors. I'll use the latter from now on. Anyway, $\vec{v} \cdot \vec{w} = \sum_{i=1}^{n} \overline{v}_i w_i$; again, notice the conjugate of v!

The norm is used to extend the idea of the length of a vector to complex, negative, and higher-dimensional vectors; the norm of a vector $\vec{v}$ (denoted $||\vec{v}||$) is $\sqrt{|v_1|^2 + |v_2|^2 + \cdots + |v_n|^2}$. If v is a complex vector, we can decompose its components into real and imaginary parts to get $\sqrt{(a_1^2 + b_1^2) + (a_2^2 + b_2^2) + \cdots + (a_n^2 + b_n^2)}$. You'll notice this is *also* $\sqrt{\overline{v_1}v_1 + \overline{v_2}v_2 + \cdots + \overline{v_n}v_n}$! This last bit is important because $\sqrt{\vec{v}^\dagger \vec{v}} = \sqrt{\vec{v} \cdot \vec{v}} = ||\vec{v}||$. Blah blah normalizing vectors $\frac{1}{||\vec{v}||}\vec{v}$ blah blah.

Gah... unit vectors are denoted with hats. $\hat{u}_1$ and such.

Linear independence, bases, etc etc. This is simple - we didn't go over determinants for linear independence or anything. Bases are a set of n linearly independent (determinant of the matrix is nonzero) vectors in $\mathbb{R}^n$ or $\mathbb{C}^n$.

We go over change of basis: expressing $\vec{x} = x_1|0\rangle + x_2|1\rangle$ in other bases by defining $|0\rangle$ and $|1\rangle$ in other bases, plugging them in, and grouping terms.

Projections! The dot product of a vector $\vec{v}$ with a **unit vector** in the direction of another vector $\vec{w}$ (so the unit vector is $\frac{\vec{w}}{||\vec{w}||}$) is the projection of $\vec{v}$ onto $\vec{w}$. This is the same as the inner product of the first vector with the projection operator of the second vector. The projection operator is the outer product of the second vector with itself, **normalized**. For example, the projection operator for $|0\rangle$ is $|0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$. Pleeeeeeease note the emphasis on normalization/unit vectors!!!

8 Day 8

Finally, quantum circuits today! Kind of hard to draw...

We start by going over projections, outer product, and inner product; the outer product also lets you write a matrix in very simple notation.

One of the matrices we'll be working with a lot is the X gate: $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, the quantum equivalent of the classical NOT gate - for example, $X|0\rangle = |1\rangle$, and vice versa. An interesting observation while we're here is that if we see what a matrix operation does on the basis vectors, we can see what it does on all vectors within the vector space of that basis, since matrices are linear operators and you can distribute them over the linear combinations of basis vectors other vectors are written in.

Other matrices/gates we'll be working with are $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$, $Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$, and $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. X, Y, Z, and I are the Pauli matrices and form our basis system for working with one qubit. X switches $|0\rangle$ and $|1\rangle$, Y changes $|0\rangle$ into $i|1\rangle$ and $|1\rangle$ into $-i|0\rangle$, Z negates $|1\rangle$, and I does nothing.

We'll also cover some basic cryptography today, since others in the class expressed interest in it.

Firstly, there is symmetric and asymmetric key cryptography. Both require a key, but symmetric uses the same key to encrypt/decrypt a message while asymmetric uses different keys for each Symmetric cryptography is faster, but asymmetric cryptography is more secure.

For instance, let K be the key and M the message. In symmetric key cryptography, $K^{-1}(KM)$ returns M, with KM being the encrypted message. In asymmetric key cryptography, $K_1(K_2M)$ returns M, with K_1 being the private key and K_2 being the public key.

There's this idea of the one-time pad: if you have a random key that's as long as the message, you can encrypt the message by XORing the key with the message; XOR is reversible, so this is an example of symmetric key cryptography. It comes from when the military would send out big books of pads, and the sender would tell the receiver which page would be used publicly, but nobody without a copy of the pad could listen in. The problem is that you need a new pad for every message, and you need to keep the pad secret. The one-time pad is provably, completely secure: without knowing the original pad, the message could theoretically decode to *anything*.

We did a little activity with making our own one-time pads and encrypting/decrypting messages with them, and that was that. Tomorrow we'll look some more into the key distribution problem and how quantum computers can tie in!

9 Day 9

We're working more with quantum gates today. We reviewed the Pauli matrices (also sometimes called unary, due to them operating on only one qubit), and got reintroduced to the Hadamard gate: $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. It's a quantum gate that takes a qubit in the $|0\rangle$ state and puts it in an equal superposition of $|0\rangle$ and $|1\rangle$. This transforms $|0\rangle$ into $|+\rangle$ and $|1\rangle$ into $|-\rangle$, and both vice versa.

We make mention of the fact that a fanout operation is impossible with quantum gates, whereas it's trivial with classical gates. It's also impossible to read the amplitudes without collapsing the wave function upon observation.

The four Pauli gates (and the Hadamard gate, at that) are not only unitary, but in fact squaring them (matrix multiplying them against themself) yields the identity matrix. This is because they're their own conjugate transposes, and thereby their own inverses (they reverse their own operations).

We talk some about quantum algorithm design: we want to, through a sequence of manipulations on input qubits, end in a result state where $|1\rangle$ indicates 100% certainty that some condition is true and $0\rangle$ indicates 100% certainty that the same condition is false.

Now we need to extend our thought to more qubits - just having one isn't particularly interesting. With two qubits, we now have two states to deal with: $|\Psi_0\rangle$ and $|\Psi_1\rangle$. You might also see the tensor product: $|\Psi_0\rangle \otimes |\Psi_1\rangle$, a way of representing 2 qubits. More commonly, you just see $|\Psi_0\Psi_1\rangle$: for instance, the 4-dimensional basis vectors are $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. This can be extended to any number of qubits! We can treat these combined states as a singular state of their own.

The tensor product is defined for two vectors (we'll give an example in 2 qubits) as follows: for $\vec{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$ and $\vec{w} = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$, $\vec{v} \otimes \vec{w} = \begin{bmatrix} v_1\vec{w} \\ v_2\vec{w} \end{bmatrix} = \begin{bmatrix} v_1 \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \\ v_2 \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} v_1w_1 \\ v_1w_2 \\ v_2w_1 \\ v_2w_2 \end{bmatrix}$. For instance, $|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$. This can be extended to any number of dimensions, and can be chained across any number of qubits in succession.

We see there are four dimensions in a 2-qubit system, with each qubit originally having 2 dimensions; now, wave functions must be written in four dimensions as well. A 2-qubit wave function can be written as $|\Psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$. Like in a 1-qubit system, this is a superposition of all basis states, with complex amplitudes, the squares of which are the probabilities of measurement.

More specifically, if the original qubits were $|\Psi_0\rangle = \alpha_0|0\rangle + \beta_0|1\rangle$ and $|\Psi_1\rangle = \alpha_1|0\rangle + \beta_1|1\rangle$, the 2-qubit system $|\Psi_{01}\rangle = |\Psi_0\rangle \otimes |\Psi_1\rangle = \alpha_0\alpha_1|00\rangle + \alpha_0\beta_1|01\rangle + \beta_0\alpha_1|10\rangle + \beta_0\beta_1|11\rangle$.

For ex., $|\Psi_0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ and $|\Psi_1\rangle = \frac{1}{2}|0\rangle + \frac{\sqrt{3}}{2}|1\rangle$, $|\Psi_{01}\rangle = \frac{1}{2\sqrt{2}}|00\rangle + \frac{\sqrt{3}}{2\sqrt{2}}|01\rangle + \frac{1}{2\sqrt{2}}|10\rangle + \frac{\sqrt{3}}{2\sqrt{2}}|11\rangle$.

We can store these four coefficients in a 4x1 vector: $\begin{bmatrix} \alpha_0\alpha_1 \\ \alpha_0\beta_1 \\ \beta_0\alpha_1 \\ \beta_0\beta_1 \end{bmatrix}$. Performing operators on this, we will still end up with a 4x1 vector; thus, the operators on a 2-qubit system must be 4x4 matrices.

We do an exercise in combining and separating 2-qubit states. However, there are some 2-qubit states that **cannot be separated into 2 1-qubit states!** For instance, $|\Psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$ cannot be separated into $|\Psi_0\Psi_1\rangle$ for any $|\Psi_0\rangle$ and $|\Psi_1\rangle$. We call these *entangled* states. For instance, the state above cannot be separated because for the $|01\rangle$ and $|10\rangle$ states to disappear, their coefficients must be zero, meaning one of the component coefficients had to have been zero in the original state - impossible, because that would also make either the $|00\rangle$ or $|11\rangle$ state disappear!

This provides a dilemma: if we measure the first qubit in an entangled state, we also gain knowledge about the state of the second qubit - in this case, the second qubit must also be 0! Collapsing either qubit also collapses the other. This is so weird that Einstein (and two other physicists) said this was BS and that quantum mechanics was incomplete, positing the EPR paradox - that entanglement vioklates locality (you can't communicate faster than the speed of light) and realism (all information about a system should be contained within that system, even if hidden). However, it's been proven that entanglement is real!

There are four "go-to" states of entanglement, called the Bell states: $|\Phi^+\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$, $|\Phi^-\rangle = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$,

$|\Psi^+\rangle = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$, $|\Psi^-\rangle = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$.

10 Day 10

We start by discussing the one-time pad again, noting that the XOR gate has an equal split of outputs regardless of the input, meaning that the output will be truly random as long as the key itself is. This guards against any possibility of an attack that tries to find a pattern in the output.

The two main problems with the one-time pad are, again, key generation and sharing. Quantum computing makes key generation much more secure, as collapsing a $|+\rangle$ state (which is what the Hadamard gate does) is truly random, as opposed to classical pseudorandom generators. In addition, entanglement can allow us to share keys more securely, and across long distances.

Let's suppose Alice and Bob are communicating over a quantum line: they each have a photon, entangled with the other's. Alice can measure her photon in any basis she wants, and Bob can measure his in any basis he wants. If they measure in the same basis, they'll get the same result; different bases, different results. This is the foundation of the BB84 protocol.

Alice's transmission basis	H	C	H	C	H	H
Alice's bit sequence	0	0	1	0	1	0
Alice's polarization	↗	↑	↖	↑	↖	↗
Bob's detection basis	C	C	C	C	H	C
Bob's bit measurement	0 or 1	0	0 or 1	0	1	0 or 1
Basis comparison	NO	YES	NO	YES	YES	NO
Sifted key		0		0	1	

H for Hadamard ($|+\rangle$, $|-\rangle$) and C for Computational ($|0\rangle$, $|1\rangle$). Both Alice and Bob pick their measurement basis randomly. After the first six measurements above, Alice sends a list of her transmission bases to Bob over an insecure classical communications channel. Bob discards any results where he measured in a basis that did not match, and he sends Alice a list of all his bases so she can do the same. The remaining bits are the sifted key, which is secure. In addition, to create a sifted key of length ℓ, they must exchange an average of 2ℓ photons.

BB84 is also secure against interference over the quantum line, since you can't copy a photon and you can only measure it once, so the interceptor would have to randomly guess what transmission basis to use when sending a new photon to Bob after intercepting Alice's. Before Alice and Bob create their sifted key, they check individual results: if they both read the same bit but in different polarizations, that means someone's listening!

We also cover some algorithms with Jacob, with a simple example being search algorithms on a sorted list: linear (O(n)) vs. binary (O(log n)). Then, we set up IBM Cloud accounts to play around with the quantum computer simulator. We get introduced to a few new gates, which we'll mathematically introduce later: for example, the SWAP gate, which just swaps two qubits' states.

We're then introduced to conditional gates: they have a condition that they test for on one qubit, and an action they perform on another qubit based on that condition. In other words, they have a control qubit and a target qubit. EVERY 1-qubit gate can be turned into a conditional 2-qubit gate. For instance, the controlled NOT (CNOT, CX) gate flips the target qubit if the control qubit is 1, and does nothing if the control is 0. However, it's a 2-qubit gate, meaning there must also be 2 outputs: the control qubit is obviously unchanged.

11 Day 11

We begin with a review of the circuit model of quantum computing and discuss how it's used to model algorithms. Again, kind of difficult to draw out. Unlike classical circuits, the "wires" don't actually represent wires, but rather the passage of quantum states over time.

We do some more review on quantum state representations, dimensions of quantum gates, and tensor product of two states; all elementary things. A good point that was brought up is that you can't always think of 2-qubit gates as 2 1-qubit gates, due to the possibility of quantum entanglement. We also go over order of operations again: writing out the gates performed on a state ket is in reverse order of the gate representation of the circuit.

Then we review linear transformations/functions/maps/operators. This is why we can describe the action of a linear operator U by defining how it behaves on the basis states: over a vector expressed in terms of the basis states as $\vec{v} = \alpha|0\rangle + \beta|1\rangle$, we separate over addition and scalar multiplication to get $U\vec{v} = \alpha U|0\rangle + \beta U|1\rangle$. It's also interesting that a 2x2 matrix operating on 1 qubit can be written with the first column being its action on $|0\rangle$ and the second its action on $|1\rangle$.

Gates, as well as the matrices associated with them, are also unitary, as we've said before. This means that $U^\dagger U = I$, or $U^{-1} = U^\dagger$. Applying them to orthonormal vectors preserves their orthonormality (and therefore their inner product), and they will always preserve angles. Note that every matrix has a conjugate transpose, but not necessarily an inverse - but our unitary operators have both. Since quantum gates are unitary, this means that anything you can do to a quantum state can always be undone, unlike in classical computing - for instance, you can't figure out the original inputs from the output of a 2-bit gate like NAND.

On the specific example of $U = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$ (letting $\begin{bmatrix} a \\ b \end{bmatrix}$ be its action on $|0\rangle$ and $\begin{bmatrix} c \\ d \end{bmatrix}$ its action on $|1\rangle$), since $U^\dagger U = \begin{bmatrix} \overline{a} & \overline{b} \\ \overline{c} & \overline{d} \end{bmatrix} \begin{bmatrix} a & c \\ b & d \end{bmatrix} = \begin{bmatrix} \overline{a}a + \overline{b}b & \overline{a}c + \overline{b}d \\ \overline{c}a + \overline{d}b & \overline{c}c + \overline{d}d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. This results in several linear equations we can set up from the last equality, importantly (and equivalently for the lower right) $\overline{a}a + \overline{b}b = 1$, which means $|a|^2 + |b|^2 = 1$ - this looks like the wave function equation! In particular, $\begin{bmatrix} a \\ b \end{bmatrix}$ is the action of U on the $|0\rangle$ state! We see the same for c and d.

In addition, the other two equations are the dot products of the two columns of U with each other, which must be 0. This means that the columns of U (which, again, are its actions on $|0\rangle$ and $|1\rangle$) are orthogonal to each other, and therefore U is unitary.

Back to reversibility from two paragraphs ago: quantum circuits need to be reversible, since we can't lose information (= losing energy) in the same way we can in classical circuits (like being unable to figure out the original inputs from the output of a NAND gate). The Bennett guy from BB84 proved that you CAN do classical computing reversibly, but it'd be inefficient to reengineer classical computers to do things reversibly to get the same results, rather than using quantum computing for new algorithms. All quantum gates are reversible, except making an observation.

We go over the outer product again and how it's useful in rewriting gates: for instance, the X gate is $|1\rangle\langle 0| + |0\rangle\langle 1|$. We can use this to simplify calculations by distributing kets: for instance, $X|1\rangle = (|1\rangle\langle 0|+|0\rangle\langle 1|)|1\rangle = |1\rangle\langle 0|1\rangle+|0\rangle\langle 1|1\rangle = |1\rangle\cdot 0+|0\rangle\cdot 1 = |0\rangle$. We can apply our understanding of the inner product as the projection of a vector (how much v is in w) to quickly calculate the inner product parts! This is interesting because outer product representations of matrices write the matrix as a sum of projection operators - the earlier representation shows the 1 state maps to 0, and vice versa; writing Z as $|0\rangle\langle 0| - |1\rangle\langle 1|$ shows the 0 state maps to 0 and the 1 state maps to -1. Y is $i|1\rangle\langle 0| - i|0\rangle\langle 1|$, and I is $|0\rangle\langle 0| + |1\rangle\langle 1|$. H is either $|+\rangle\langle 0| + |-\rangle\langle 1|$ or $|0\rangle\langle +| + |1\rangle\langle -|$.

The last 1-qubit gate is the phase gate, representing a rotation by an angle ϕ: it does nothing to $|0\rangle$ but rotates $|1\rangle$ by ϕ: $R_\phi|1\rangle = e^{i\phi}|1\rangle$. In fact, the Z gate is a special case of the phase gate where $\phi = \pi$. The phase gate is $|0\rangle\langle 0| + e^{i\phi}|1\rangle\langle 1|$.

We then go over the CNOT gate again, which is more complicated because it's 4x4, a 2-qubit gate. It's defined as $|00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 11| + |11\rangle\langle 10|$. Again, each bra maps to the respective ket. We can also write it as $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X$ - applying this to a state $|q_1 q_2\rangle$, this says: "If q_1 is 0, map q_1 to 0 and apply the identity matrix to the second qubit (since the second term cancels to 0). If q_1 is 1, map q_1 to 1 and apply the X (NOT) gate to the second qubit (since the first term cancels to 0)."

Note that the tensor product doesn't distribute the way you'd expect it to: $(M_1 \otimes M_2)(|q_1\rangle \otimes |q_2\rangle) = M_1|q_1\rangle \otimes M_2|q_2\rangle$. Evaluating each term and tensoring them together at the end gives the same result as calculating it out, but in a simpler fashion.

12 Day 12

We're starting right off with the CNOT gate again today, with its action described with outer products as shown yesterday. Today, we begin with the CZ - controlled Z - gate, which operates conditionally like the CNOT/CX. Since the CZ gate only operates on the $|1\rangle$ state, we can describe it as $CZ = |00\rangle\langle 00| + |01\rangle\langle 01| + |10\rangle\langle 10| - |11\rangle\langle 11|$. This is because the $|11\rangle$ state is negated, while the other states are unchanged. We can also write it as $|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Z$.

There's also the CY, CPhase, and other gates. There are also 3-qubit gates, like the Fredkin gate, which is basically a conditional SWAP gate. It's a bit annoying to write out all 8 possible states, so we number them $|0\rangle$ to $|7\rangle$. There's also the Toffoli gate, which requires two inputs to invert the third: a CCX gate. Writing it out in outer products takes a while, and the 8x8 matrix is even

worse!

With all this knowledge about outer products, we are now equipped to explain the gate we were left with on Friday, with the top qubit running through a Hadamard gate and then into the control qubit of a CX gate. Both starting at $|0\rangle$, the total state after the H gate is $|\Psi_1\rangle = |+\rangle|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|0\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|10\rangle$. Running this through a CNOT gate, this becomes $\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. This is a Bell state, and is entangled!

(I'm not paying much attention in class right now, so the following notes are weird.)

We then go over the copy gate - or, rather, why it can't exist! The hypothetical copy gate U_{copy} copies the first qubit onto the second in a 2-qubit system. For instance, copying a state $|\Psi\rangle|0\rangle$ is $U_{copy}(\alpha|0\rangle + \beta|1\rangle)|0\rangle = \alpha U_{copy}|00\rangle + \beta U_{copy}|10\rangle = \alpha|00\rangle + \beta|11\rangle$, which is an entangled state! However, applying the gate over both states first, rather than writing $|\Psi\rangle$ in the computational basis, just makes it separable - it becomes $|\Psi\rangle|\Psi\rangle$! There are more ways to prove this can't be possible, like expanding out probability amplitudes.

We go over Deustch's problem. Remember that there are 4 1-bit classical operations: force it to 0 or 1, NOT, and nothing. We have a one-bit circuit, and an "oracle" on it - this oracle is a hidden gate, where we can pass in an input (query it) and get an unknown output, but we know it's one of the 4 classical operations. In classical computing, we need 2 queries to figure out whether the gate is constant (always one output) or balanced (not/nothing) (and for that matter, we know exactly which gate it is). These classical gates are not reversible, either, so we cannot represent them as unitary matrices and therefore quantum gates! Our quantum equivalent uses 2 inputs and 2 outputs to make it reversible: we have inputs $|X\rangle$ and $|0\rangle$, but we keep $|X\rangle$ and write the result to the auxiliary qubit $|0\rangle$ - specifically, the state becomes $|0 \oplus f(x)\rangle$.

If we make $|X\rangle$ a $|+\rangle$, we see funny things happen. The operation of the gate U_f on $|X\rangle|0\rangle$ is $|X\rangle|0 \oplus f(X)\rangle$, so just before that $|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$. We can now apply our U_f onto that to get $|\Psi'\rangle = \frac{1}{\sqrt{2}}(U_f|00\rangle + U_f|10\rangle) = \frac{1}{\sqrt{2}}(|0 \oplus f(0)\rangle|0\rangle + |0 \oplus f(1)\rangle|1\rangle) = \frac{1}{\sqrt{2}}(|f(0)\rangle|0\rangle + |f(1)\rangle|1\rangle)$. We see we evaluate *both* $f(0)$ and $f(1)$ in one query to the oracle! However, we cannot do multiple queries in one query like this, because we need to be able to measure them separately. This paves the way to Deutsche's algorithm, which is the first actual application of quantum computing!

We can expand the above into higher dimensions of $|X\rangle$, and with still a single auxiliary qubit for output, the output is still $|f(X)\rangle$. If, rather than $|X\rangle$, we send in the $|+\rangle$ state in each input qubit, we get every single possible output in the output qubit! This concept is known as quantum parallelism: we can evaluate multiple inputs at once, but we can't get the outputs out.

Now we build Deutsch's quantum circuit. It starts with a 0 and a 1, both passed through H gates, and thrown into an oracle U_f with the former 1 (now -) being the bit to be copied to. The 0 (now +), however, is of interest: it is unchanged by U_f, but is passed through a H gate again and finally measured. The initial state after the first H gates but before the oracle is $|\Psi_1\rangle = |+-\rangle$, then after the gate it becomes $|\Psi_2\rangle = U_f \cdot \frac{1}{2}(|00\rangle + |10\rangle - |01\rangle - |11\rangle)$, and U_f can be distributed across each. This becomes $|\Psi_2\rangle = \frac{1}{2}(|0\rangle|f(0)\rangle + |1\rangle|f(0)\rangle - |0\rangle|1 \oplus f(1)\rangle - |1\rangle|1 \oplus f(1)\rangle)$. We can now apply the H gate to the first qubit again, and we get $|\Psi_3\rangle = \frac{1}{2}(|0f(0)\rangle + |1f(0)\rangle - |0f(1)\rangle - |1f(1)\rangle) =$

$\frac{1}{2}(|0\rangle+|1\rangle)(|f(0)\rangle-|f(1)\rangle)$. We can now measure the first qubit, and we see that if it's 0, $f(0) = f(1)$ (working out the math reveals that the state is separable, it's the original state), and if it's 1, $f(0) \neq f(1)$! This is a single query to the oracle that shows whether or not the gate is constant (if the original bit is 0), and we can't do this in classical computing.

13 Day 13

Today we're covering Grover's and Shor's algorithms to start. Grover's algorithm (1996) is a slightly faster quantum search algorithm, while Shor's algorithm (1994) is an enormous optimization for the prime factorization problem.

We talk about the NAND gate some more, and bring up the Toffoli (CCX) gate with classical values. Not sure why. Then it's onto basic classical search algorithms - binary search for numbers. Then, we talk about searching for particular words in a string - linear search is the best you can do classically. It's like a sorted list vs. unsorted list.

Anyway, onto Grover's algorithm. Grover's algorithm lets us search an unsorted list in $O(\sqrt{N})$, rather than the classical $O(N)$. We can describe our search space as a vector $|X\rangle$ with coefficients, as yesterday, $\frac{1}{\sqrt{2^n}}$, over an equal superposition of every element: $|0\rangle + |1\rangle + |2\rangle + \cdots + |n\rangle$. We can recreate this in quantum computing using using, like yesterday, the H gate: run each of our input qubits through a H gate and through an oracle, and we get an equal superposition of all our elements, which we can use to compare *everything* at once against what we're looking for.

Let's say our oracle's function $f(x)$ returns 1 if $x = x*$, the value we're looking for, and 0 otherwise. This can be mathematically represented as $O|\Psi\rangle = \sum_x \alpha_x(-1)^{f(x)}|x\rangle$ - the operation of O on a state $|\Psi\rangle$. α_x is the coefficient that was on the $|X\rangle$ state before; the oracle doesn't change the coefficient here. Expanding this as outer products, this is $O = |0\rangle\langle 0| + |1\rangle\langle 1| + \cdots - |4\rangle\langle 4| + \cdots + |n\rangle\langle n|$, if we're looking for $|4\rangle$.

Operating on the original state from two paragraphs ago, this just flips the sign on the state being searched for; however, we still need to read that answer. We didn't get that far before Alan had to go to a meeting, and we did another activity with the IBM simulator.

We walk through the methods of superposing two qubits using a H and CX gate, and talk about the difference between experimental and theoretical results: the IBM results aren't exactly 50/50 in experiment when they should be in theory, due to a variety of noise issues and relatively low (1024 shots) sample size. We also see the actual matrix form of the CX gate: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$.

And Alan is back and we're back to Grover's algorithm. Suppose we have n states of $|0\rangle$, written as $|\Psi_0\rangle = |0\rangle^{\otimes n}$; then, all are sent through H gates, resulting in $|\Psi_1\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + |1\rangle + \cdots + |n\rangle)$. Then, applying the oracle, $O|\Psi\rangle = \sum i\alpha_i(-1)^{f(x)}|i\rangle$, not to be confused with the imaginary unit. Again, just reiterating what we said earlier. We do some more math, but honestly I'm kind of lost here, so you can just Google it. Apparently, trying to invert the now-negative amplitude around the

mean of amplitudes amplifies the amplitude of what we're looking for and reduces the amplitudes of everything else; repeatedly running it through the oracle and inverting over the mean eventually makes the amplitude of what we're looking for almost 1 and everything else almost 0, and we can measure it with high accuracy after $\sqrt{n}$ attempts.

14 Day 14

We're right into Shor's algorithm today: finding prime factors! The classical way of doing this, factoring a semi-prime P into the product of two distinct primes Q, R is as follows:

1. Choose a random integer a
2. Calculate $a^x \text{mod} P$ where $x = 0, 1, 2, 3, \ldots$
3. Find the periodicity r of the above sequence
4. If r is odd, start over; otherwise, $Q = \gcd(a^{\frac{r}{2}} - 1, P)$ and $R = \gcd(a^{\frac{r}{2}} + 1, P)$

Finding periodicity is the hard part for classical computers.

I was working on something else and we're on modular arithmetic now. We're going over step 2 above: $a^x \text{mod} P$. We look at $a = 7$, $P = 15$: you see a periodicity in the sequence every 4 exponents: 1, 7, 4, 13, 1, 7, 4, 13...

We refer back to quantum registers: a group of qubits storing a particular value. Now, we set up our oracle system again, but with two n-qubit registers going in to an oracle that evaluates $a^x \text{mod} P$. However, since the modulo operation is not reversible (3 mod 12 is the same as 15 mod 12), we again use an auxiliary, but now we use an auxiliary *register*. The top register is $|x\rangle$, all passed through Hadamard gates in typical fashion, and the lower one is $|\Psi_0\rangle$; the auxiliary register preserves $|x\rangle$ and the lower one outputs $|0\rangle|a^x \text{mod} P\rangle$.

The original state before the Hadamard gates is $|0000\ldots0\rangle|0000\ldots0\rangle$, then after is $|+++\cdots+\rangle|0000\ldots0\rangle$. We can write the total state here $|\Psi_1\rangle$ as $\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|0\rangle$. After passing through the oracle, this now becomes $|\Psi_2\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle|a^x \text{mod} P\rangle$.

Measuring only the 2nd register, the one with the interesting output, we will get a number between 0 and $P-1$: it is written as $\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|a^x \text{mod} P\rangle$. However, due to some sort of light entanglement going on inside the oracle, the 1st register gets reduced when we measure the 2nd register. In fact, $|\Psi_3\rangle = \frac{1}{\sqrt{m}}\sum_{K=0}^{m-1}|x_0 + Kr\rangle|M\rangle$, where M is the collapsed state of register 2 and m is the number of states remaining. x_0 is the smallest x such that $a^x \text{mod} P = M$. (This is really confusing. Maybe I shouldn't have zoned out.)

For instance, back to the examples of 7 with a remainder 4, we get $\frac{1}{\sqrt{m}}(|2\rangle|4\rangle + |6\rangle|4\rangle + |10\rangle|4\rangle + |14\rangle|4\rangle)$. Somehow this turns into the Fourier transform, I dunno.

Well, that's that, and we've got a few guest lecturers to round out the last 45 minutes of class.

We'll also be going to eat lunch with all of them later. The first one goes over what university physics is like, and the second went over quantum chemistry.

15 Day 15

Last day; it was just presentations the whole time. That'll be it for this notes document; thanks for reading!