CPEN 400Q Lecture 04
Entanglement and multi-qubit systems

Wednesday 17 January 2024

Announcements

- Technical assignment 1 out by end of week
- Quiz 2 on Monday

We introduced the "bra" part of the "bra-ket notation"

$$|v\rangle = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad \langle v| = \left(|v\rangle \right)^\dagger = \begin{pmatrix} v_1^* & v_2^* \end{pmatrix}$$

The inner product between two states is defined as

$$\langle v | w \rangle = \langle v, w \rangle = v_1^* w_1 + v_2^* w_2$$

Inner product tells about the *overlap* (similarity) between states.

orthogonal

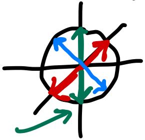
$$0 \le |\langle v | w \rangle|^2 \le 1$$

same state

Last time

We introduced the concept of *orthonormal bases* for qubit states:

$$\{|\psi_1\rangle, |\psi_2\rangle\} \rightarrow \langle\psi_i|\psi_j\rangle = \begin{cases} 1 & i=j \\ 0 & i\neq j \end{cases}$$



Examples:

$$\{|0\rangle, |1\rangle\}$$

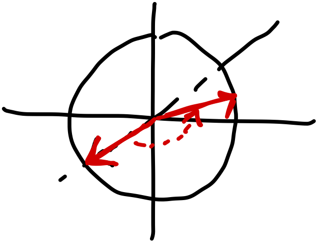
$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

$$|p\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle) \quad |m\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$$

Last time

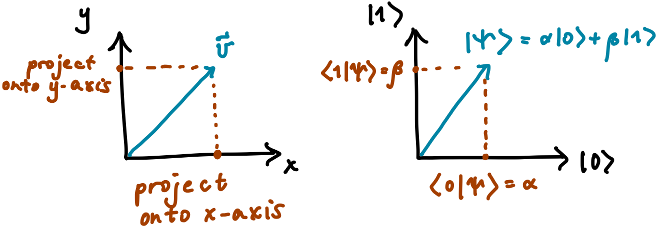
$$e^{i\phi}\left(|0\rangle + e^{i\varphi}|1\rangle\right)$$

We distinguished between global phase and relative phase.



$$|0\rangle + e^{i\phi}|1\rangle$$

We discussed *projective measurement* with respect to a basis.



When we measure state $|\varphi\rangle$ with respect to basis $\{|\psi_i\rangle\}$, the probability of obtaining outcome i is

$$\Pr(\text{outcome } i) = |\langle \psi_i | \varphi \rangle|^2$$

Image credit: Xanadu Quantum Codebook I.9

Learning outcomes

- Measure a qubit in a different basis
- Mathematically describe a system of multiple qubits
- Describe the action of common multi-qubit gates
- Make any gate a controlled gate
- Perform measurements on multiple qubits

Basis rotations

Use a basis rotation to "trick" the quantum computer.

Suppose we want to measure in the "Y" basis:

$$|p\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + i|1\rangle\right), \quad |m\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - i|1\rangle\right).$$

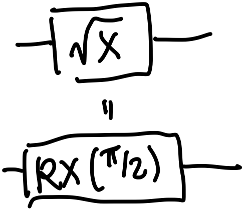
Unitary operations preserve length *and* angles between normalized quantum state vectors. (**Exercise:** prove it!)

There exists some unitary transformation that will convert between this basis and the computational basis.

Basis rotations

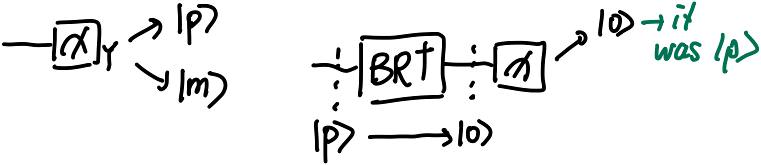
Exercise: determine a quantum circuit that sends

$$
\begin{aligned}
|0\rangle \quad &\rightarrow \quad |p\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + i|1\rangle\right) \\
|1\rangle \quad &\rightarrow \quad |m\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - i|1\rangle\right)
\end{aligned}
$$



Basis rotations

At the end of our circuit, we can then apply the reverse (adjoint) of this transformation rotate *back* to the computational basis.



That way, if we measure and observe $|0\rangle$, we know that this was previously $|p\rangle$ in the Y basis (and similarly for $|m\rangle$).

Adjoints

In PennyLane, we can compute adjoints of operations *and* entire quantum functions using `qml.adjoint`:

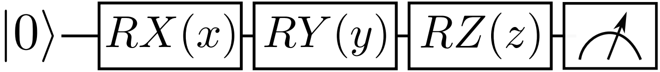
```
def some_function(x):
    qml.RZ(Z, wires=0)

def apply_adjoint(x):
    qml.adjoint(qml.S)(wires=0)
    qml.adjoint(some_function)(x)
```

`qml.adjoint` is a special type of function called a **transform**. We will cover transforms in more detail later in the course.

Basis rotations: hands-on

Let's run the following circuit, and measure in the Y basis



Hands-on time...

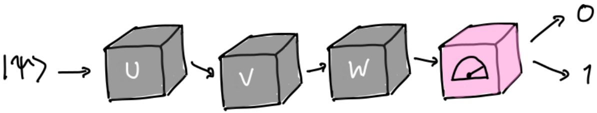
Multi-qubit systems

Recall this slide from lecture 1...

Quantum computing

Quantum computing is the act of manipulating the state of qubits in a way that represents solution of a computational problem:

1. **Prepare** qubits in a **superposition**
2. Apply **operations** that **entangle** the qubits and manipulate the amplitudes
3. **Measure** qubits to extract an answer
4. Profit



Let's simulate this using NumPy.

How do we express the mathematical space of multiple qubits?

Hilbert spaces compose under the *tensor product*.

Let

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, \quad B = \begin{pmatrix} e & f \\ g & h \end{pmatrix}.$$

The tensor product of A and B, $A \otimes B$ is

$$A \otimes B = \begin{pmatrix} a\begin{pmatrix} e & f \\ g & h \end{pmatrix} & b\begin{pmatrix} e & f \\ g & h \end{pmatrix} \\ c\begin{pmatrix} e & f \\ g & h \end{pmatrix} & d\begin{pmatrix} e & f \\ g & h \end{pmatrix} \end{pmatrix} = \begin{pmatrix} ae & af & be & bf \\ ag & ah & bg & bh \\ ce & cf & de & df \\ cg & ch & dg & dh \end{pmatrix}$$

$|0\rangle$, $|1\rangle$

Qubit state vectors are also combined using the *tensor product*:

$$|01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix}1\\0\end{pmatrix} \otimes \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1\begin{pmatrix}0\\1\end{pmatrix}\\0\begin{pmatrix}0\\1\end{pmatrix}\end{pmatrix} = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}$$

An n-qubit state is therefore a vector of length 2^n.

$|00101100\ldots\rangle = |0\rangle \otimes |0\rangle \otimes |1\rangle \otimes \ldots$

The states $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ are the computational basis vectors for 2 qubits:

$$|00\rangle = \begin{pmatrix}1\\0\\0\\0\end{pmatrix}, \quad |01\rangle = \begin{pmatrix}0\\1\\0\\0\end{pmatrix}, \quad |10\rangle = \begin{pmatrix}0\\0\\1\\0\end{pmatrix}, \quad |11\rangle = \begin{pmatrix}0\\0\\0\\1\end{pmatrix}$$

We can create arbitrary linear combinations of them as long as the normalization on the coefficients holds.

Same pattern for 3 qubits: $|000\rangle, |001\rangle, \ldots, |111\rangle$.

$$\alpha_0|000\rangle + \alpha_1|001\rangle + \ldots + \alpha_7|111\rangle \Rightarrow \sum_i |\alpha_i|^2 = 1$$

The tensor product is linear and distributive, so if we have

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\varphi\rangle = \gamma|0\rangle + \delta|1\rangle,$$

then they tensor together to form

$$|\psi\rangle \otimes |\varphi\rangle = \left(\alpha|0\rangle + \beta|1\rangle\right) \otimes \left(\gamma|0\rangle + \delta|1\rangle\right)$$
$$= \alpha\gamma|00\rangle + \beta\gamma|10\rangle + \alpha\delta|01\rangle + \beta\delta|11\rangle$$

Multi-qubit systems

Single-qubit unitary operations also compose under tensor product.

For example, apply U_1 to qubit $|\psi\rangle$ and U_2 to qubit $|\varphi\rangle$:

$$(U_1 \otimes U_2)|\psi\rangle \otimes |\varphi\rangle = U_1|\psi\rangle \otimes U_2|\varphi\rangle$$

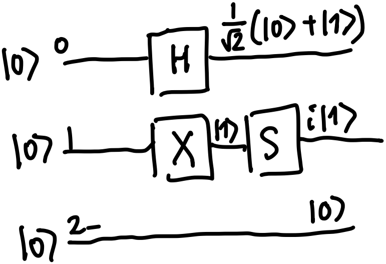
If an n-qubit ket is a vector with length 2^n, then a unitary acting on n qubits has dimension $2^n \times 2^n$.

ans: $i|+10\rangle$

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$$

Exercise: determine the state of a 3-qubit system if H is applied to qubit 0, X and then S are applied to qubit 1.

start from $|000\rangle$

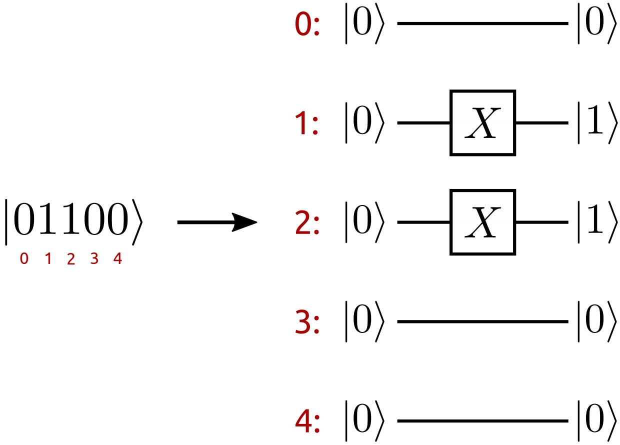


$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes i|1\rangle \otimes |0\rangle$$

$$= \frac{i}{\sqrt{2}}\left(|010\rangle + |110\rangle\right)$$

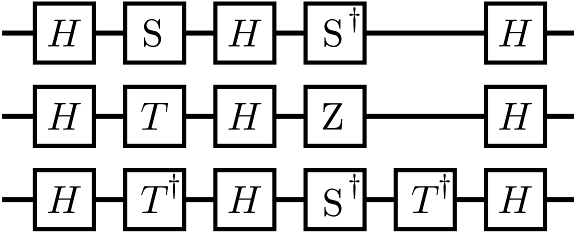
Qubit ordering (very important!)

In PennyLane:



(This is different in other frameworks!)

The few small circuits we've seen so far only involve gates on single qubits:



Surely this isn't all we can do...

Image credit: Xanadu Quantum Codebook I.11

SWAP

We can swap the state of two qubits using the SWAP operation.
First define what it does to the basis states...

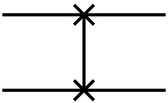
$$SWAP|00\rangle = |00\rangle$$
$$SWAP|01\rangle = |10\rangle$$
$$SWAP|10\rangle = |01\rangle$$
$$SWAP|11\rangle = |11\rangle$$

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Try it!

$(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle)$

Circuit element:



PennyLane: `qml.SWAP`

CNOT

Consider a two-qubit operation U with the following action on the basis states:

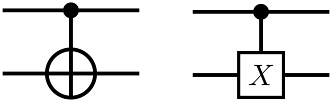
$$
\begin{aligned}
U|00\rangle &= |00\rangle \\
U|01\rangle &= |01\rangle \\
U|10\rangle &= |11\rangle \\
U|11\rangle &= |10\rangle
\end{aligned}
$$

CNOT

CNOT = "controlled-NOT". A NOT (X) is applied to second qubit only if first qubit is in state $|1\rangle$.

$$
\begin{aligned}
CNOT|00\rangle &= |00\rangle \\
CNOT|01\rangle &= |01\rangle \\
CNOT|10\rangle &= |11\rangle \\
CNOT|11\rangle &= |10\rangle
\end{aligned}
\qquad
CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}
$$

Circuit elements:



PennyLane: `qml.CNOT`

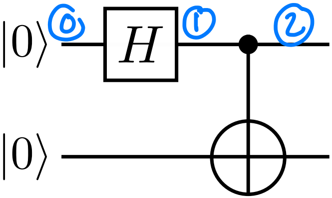
Image credit: Xanadu Quantum Codebook I.12

CNOT hands-on

$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$

Exercise:

What does CNOT do with qubits in a superposition?

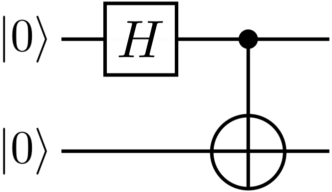


0 $|00\rangle$

1 $\left(\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)\right) \otimes |0\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle+|10\rangle\right)$

2 $\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$ Bell state

CNOT hands-on



The output state of this circuit is:

$$CNOT \cdot (H \otimes I) |00\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$$

This state is **entangled**, and CNOT is an **entangling gate**!

Entanglement

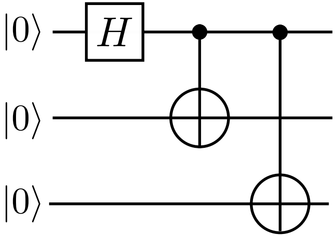
$$(\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle)$$

We *cannot express*

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right)$$

as a tensor product of two single-qubit states.

Entanglement generalizes to more than two qubits:



Exercise: Express the output state of this circuit in the computational basis.

Consider the AND of two bits a and b:

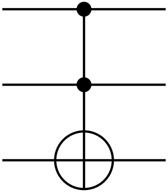
a	b	ab
0	0	0
0	1	0
1	0	0
1	1	1

This gate is *not* reversible: we cannot recover inputs from outputs.

But, we can make it reversible by adding one extra bit...

Toffoli

The **Toffoli** implements a reversible AND gate. (It is universal for classical reversible computing).



Controlled-CNOT, or controlled-controlled-NOT.

PennyLane: `qml.Toffoli`

Toffoli

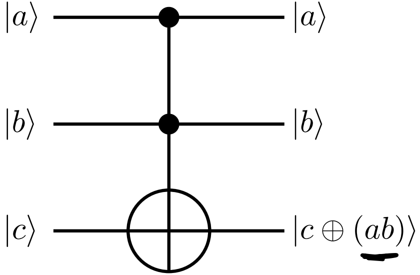
What does it do to the basis states?

$$
\begin{aligned}
TOF|000\rangle &= |000\rangle \\
TOF|001\rangle &= |001\rangle \\
TOF|010\rangle &= |010\rangle \\
TOF|011\rangle &= |011\rangle \\
TOF|100\rangle &= |100\rangle \\
TOF|101\rangle &= |101\rangle \\
TOF|110\rangle &= |111\rangle \\
TOF|111\rangle &= |110\rangle
\end{aligned}
$$

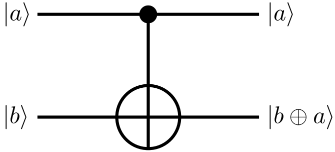
Toffoli

What is actually going on here?

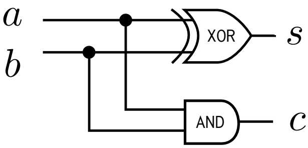
$$
\begin{aligned}
TOF|000\rangle &= |000\rangle \\
TOF|001\rangle &= |001\rangle \\
TOF|010\rangle &= |010\rangle \\
TOF|011\rangle &= |011\rangle \\
TOF|100\rangle &= |100\rangle \\
TOF|101\rangle &= |101\rangle \\
TOF|110\rangle &= |111\rangle \\
TOF|111\rangle &= |110\rangle
\end{aligned}
$$

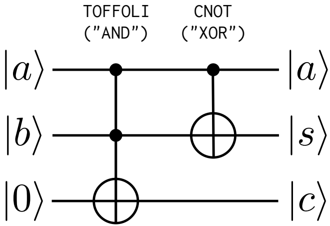


We can interpret CNOT in a similar way.



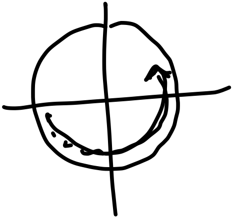
X, CNOT, TOF can be used to create Boolean arithmetic circuits.

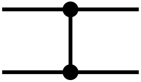


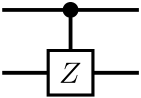


Example: controlled-Z (CZ)

What does this operation do?







PennyLane: `qml.CZ`

$|00\rangle \rightarrow |00\rangle$

$|01\rangle \rightarrow |01\rangle$

$|10\rangle \rightarrow |10\rangle$

$|11\rangle \rightarrow -|11\rangle$

$|00\rangle \rightarrow |00\rangle$

$|01\rangle \rightarrow |01\rangle$

$|10\rangle \rightarrow |1+\rangle$

$|11\rangle \rightarrow |1-\rangle$

controlled Hadamard

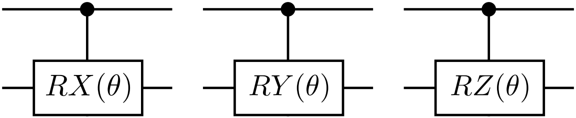
Image credit: Codebook node I.13

Example: controlled rotations (RX, RY, RZ)

Or this one?

$$CRY(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ 0 & 0 & \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

Circuit elements:



PennyLane: `qml.CRX`, `qml.CRY`, `qml.CRZ`

Image credit: Codebook node I.14

There is a pattern here:

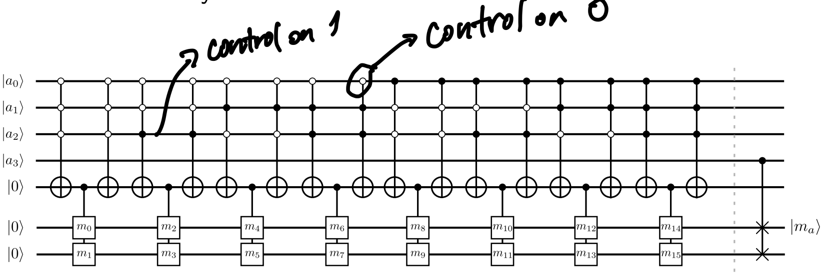
$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad CRY(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ 0 & 0 & \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{pmatrix}$$

More generally,

$$CU = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & U_{00} & U_{01} \\ 0 & 0 & U_{10} & U_{11} \end{pmatrix} = \begin{pmatrix} I_2 & \mathbf{0}_2 \\ \mathbf{0}_2 & U \end{pmatrix}$$

... we don't want to be writing these matrices all the time.

Any unitary operation can be turned into a controlled operation, controlled on any state.



Most common controls are controlled-on-$|1\rangle$ (filled circle), and controlled-on-$|0\rangle$ (empty circle).

Hands-on: `qml.ctrl`

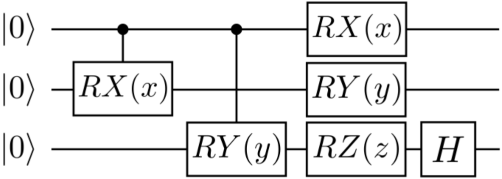
Remember from earlier, `qml.adjoint`:

```
@qml.qnode(dev)
def my_circuit():
    qml.S(wires=0)
    qml.adjoint(qml.S)(wires=0)
    return qml.sample()
```

There is a similar *transform* that allows us to perform arbitrary controlled operations (or entire quantum functions)!

```
@qml.qnode(dev)
def my_circuit():
    qml.S(wires=0)
    qml.ctrl(qml.S, control=1)(wires=0)
    return qml.sample()
```

Let's go implement this circuit:



Review: single-qubit measurements

Given a state

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

- the probability of measuring and observing the qubit in state $|0\rangle$ is $|\alpha|^2 = \alpha\alpha^* = |\langle 0|\psi\rangle|^2$
- the probability of measuring and observing the qubit in state $|1\rangle$ is $|\beta|^2 = |\langle 1|\psi\rangle|^2$
- we can measure in different bases by "remapping" those basis states to the computational basis

We can do all this in the multi-qubit case as well.

Multi-qubit measurement outcome probabilities

Let

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$$

If we measure in the computational basis, the outcome probabilities are:

- $|\alpha|^2 = |\langle 00|\psi\rangle|^2$ for $|00\rangle$
- $|\beta|^2 = |\langle 01|\psi\rangle|^2$ for $|01\rangle$
- ...

Multi-qubit measurement outcome probabilities

Let

$$|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$$

We can also measure *just one qubit*:

- The probability of the first qubit being in state $|0\rangle$ is $|\alpha|^2 + |\beta|^2$
- The probability of the second qubit being in state $|1\rangle$ is $|\beta|^2 + |\delta|^2$

Recap

- Measure a qubit in a different basis
- Mathematically describe a system of multiple qubits
- Describe the action of common multi-qubit gates
- Make any gate a controlled gate
- Perform measurements on multiple qubits

Next time

Content:

- Superdense coding
- The no-cloning theorem
- Quantum teleportation

Action items:

1. Technical assignment 1 posted this week
2. Quiz 2 Monday about contents from this week

Recommended reading:

- From this class: Codebook nodes I.9, I.11-I.14
- For next class: Codebook nodes I.15