Quantum Algorithms

Lecture: Joran van Apeldoorn
Exercises: Arjan Cornelissen

CWI

9th of September

QuSoft
Research Center for Quantum Software

Quantum algorithms

- There are many quantum algorithms.

Quantum algorithms

- There are many quantum algorithms.
- There are even more misconceptions about quantum algorithms.

Quantum algorithms

- There are many quantum algorithms.
- There are even more misconceptions about quantum algorithms.
- Goal: Get an understanding of how quantum algorithms work

Quantum algorithms

- There are many quantum algorithms.
- There are even more misconceptions about quantum algorithms.
- Goal: Get an understanding of how quantum algorithms work and how they do not work.

Quantum algorithms

- There are many quantum algorithms.
- There are even more misconceptions about quantum algorithms.
- Goal: Get an understanding of how quantum algorithms work <u>and</u> how they do not work.
- We will focus on the techniques and insight.
  (The not so famous, but really cool stuff)

Overview

- Basics - Quantum states, gates and circuits.

Overview

- Basics - Quantum states, gates and circuits.
- Amplitude amplification - Picking out solutions.

Overview

- Basics - Quantum states, gates and circuits.
- Amplitude amplification - Picking out solutions.
- Phase estimation - Investigating unitaries.

Overview

- Basics - Quantum states, gates and circuits.
- Amplitude amplification - Picking out solutions.
- Phase estimation - Investigating unitaries.
- Amplitude estimation - Estimating probabilities.

Overview

- Basics - Quantum states, gates and circuits.
- Amplitude amplification - Picking out solutions.
- Phase estimation - Investigating unitaries.
- Amplitude estimation - Estimating probabilities.
- Overview of other algorithms
    - Shors factoring algorithm.
    - HHL for systems of linear equations.
    - Gradient computation.

Overview

- Basics - Quantum states, gates and circuits.
- Amplitude amplification - Picking out solutions.
- Phase estimation - Investigating unitaries.
- Amplitude estimation - Estimating probabilities.
- Overview of other algorithms
  - Shors factoring algorithm.
  - HHL for systems of linear equations.
  - Gradient computation.
- A quick quiz!

Basics

Quantum states

- A bit is 0 or 1, a qubit is in a superposition of $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

Quantum states

- A bit is 0 or 1, a qubit is in a superposition of $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

- If we measure then we get one outcome.
  The probability of measuring $|0\rangle$ is $|\alpha_0|^2$.
  The probability of measuring $|1\rangle$ is $|\alpha_1|^2$.

Quantum states

- A bit is 0 or 1, a qubit is in a superposition of $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

- If we measure then we get one outcome.
  The probability of measuring $|0\rangle$ is $|\alpha_0|^2$.
  The probability of measuring $|1\rangle$ is $|\alpha_1|^2$.
- We combine qubits to create bigger states via tensor products.

Quantum states

- A bit is 0 or 1, a qubit is in a superposition of $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

- If we measure then we get one outcome.
  The probability of measuring $|0\rangle$ is $|\alpha_0|^2$.
  The probability of measuring $|1\rangle$ is $|\alpha_1|^2$.
- We combine qubits to create bigger states via tensor products.
- Quantum states are normalized complex vectors, the classical states $|0\rangle, |1\rangle, |2\rangle, \ldots$ form a basis. (why normalized?)

Quantum states

- A bit is 0 or 1, a qubit is in a superposition of $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$$

- If we measure then we get one outcome.
  The probability of measuring $|0\rangle$ is $|\alpha_0|^2$.
  The probability of measuring $|1\rangle$ is $|\alpha_1|^2$.
- We combine qubits to create bigger states via tensor products.
- Quantum states are normalized complex vectors, the classical states $|0\rangle, |1\rangle, |2\rangle, \ldots$ form a basis. (why normalized?)
- For a qubit:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.
- Unitaries on only a few qubits are called gates.

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.
- Unitaries on only a few qubits are called gates.

$$I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.
- Unitaries on only a few qubits are called <u>gates</u>.

$$I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

- I does nothing.

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.
- Unitaries on only a few qubits are called gates.

$$I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

- I does nothing.
- X changes $|0\rangle$ into $|1\rangle$ and vice versa.

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.
- Unitaries on only a few qubits are called gates.

$$I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

- I does nothing.
- X changes $|0\rangle$ into $|1\rangle$ and vice versa.
- Z adds a -1 in front of $|1\rangle$.

Quantum gates

- We can change states by applying unitaries, since they keep vectors normalized.
- Unitaries on only a few qubits are called gates.

$$I := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

- I does nothing.
- X changes $|0\rangle$ into $|1\rangle$ and vice versa.
- Z adds a -1 in front of $|1\rangle$.
- H changes $|0\rangle$ and $|1\rangle$ into $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$.

Changing the basis

Claim: H is a basis transform

Changing the basis

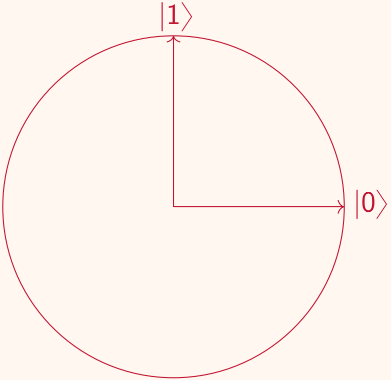
Claim: H is a basis transform

$$|+\rangle := \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad |-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Changing the basis

Claim: H is a basis transform

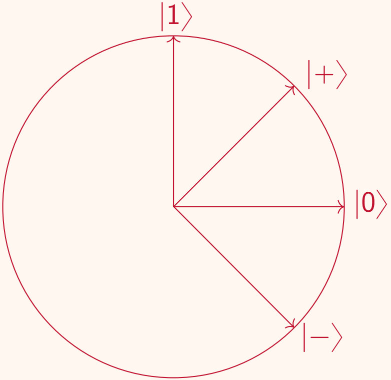
$$|+\rangle := \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad |-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$



Changing the basis

Claim: H is a basis transform

$$|+\rangle := \frac{|0\rangle + |1\rangle}{\sqrt{2}} \qquad |-\rangle := \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$



A first circuit

What does this circuit do? We only need to try a basis (Why?).

A first circuit

$$-\boxed{H}-\boxed{Z}-\boxed{H}-$$

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

A first circuit

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

$$|0\rangle \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

$$-\boxed{H}-\boxed{Z}-\boxed{H}-$$

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

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

A first circuit

$$-\boxed{H}-\boxed{Z}-\boxed{H}-$$

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

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

A first circuit

$$
-\boxed{H}-\boxed{Z}-\boxed{H}-
$$

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

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

- On $|1\rangle$:

$$
|1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}}
$$

A first circuit

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

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

- On $|1\rangle$:

$$|1\rangle \mapsto \frac{|0\rangle - |1\rangle}{\sqrt{2}} \mapsto \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

A first circuit

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

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

- On $|1\rangle$:

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

A first circuit

$$-\boxed{H}-\boxed{Z}-\boxed{H}-$$

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

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

- On $|1\rangle$:

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

This is a X gate!

A first circuit

What does this circuit do? We only need to try a basis (Why?).

- On $|0\rangle$:

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

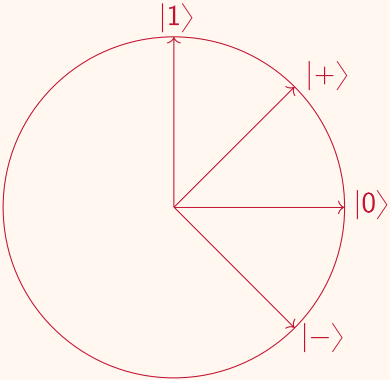
- On $|1\rangle$:

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

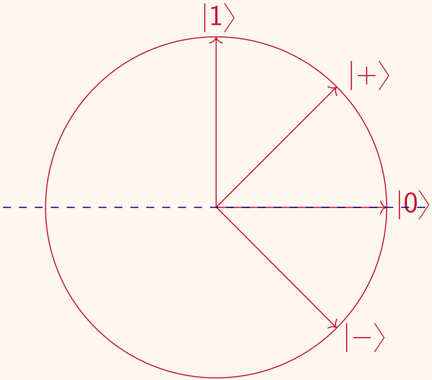
This is a X gate! Z is just X in the $\{|+\rangle, |-\rangle\}$ basis (and vice versa).

Reflections

We can also see this in our image.

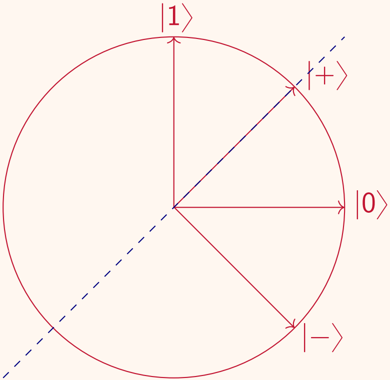


We can also see this in our image.
Z is a reflection through the $|0\rangle$ state.



Reflections

We can also see this in our image.
Z is a reflection through the $|0\rangle$ state.
X is a reflection through the $|+\rangle$ state.



More gates

- A phase is a factor of the form $e^{i2\pi\theta}$.

More gates

- A phase is a factor of the form $e^{i2\pi\theta}$.

$$\text{---}\boxed{e^{i2\pi\theta} I}\text{---} = \begin{bmatrix} e^{i2\pi\theta} & 0 \\ 0 & e^{i2\pi\theta} \end{bmatrix}$$

More gates

- A phase is a factor of the form $e^{i2\pi\theta}$.

$$\boxed{e^{i2\pi\theta} I} = \begin{bmatrix} e^{i2\pi\theta} & 0 \\ 0 & e^{i2\pi\theta} \end{bmatrix}$$

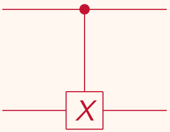
$$\boxed{\pi/2} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix}$$

- A phase is a factor of the form $e^{i2\pi\theta}$.

$$\boxed{e^{i2\pi\theta} I} = \begin{bmatrix} e^{i2\pi\theta} & 0 \\ 0 & e^{i2\pi\theta} \end{bmatrix}$$

$$\boxed{\pi/2} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix}$$

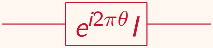
- We can also make gates controlled:



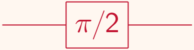
$=$

More gates

- A <u>phase</u> is a factor of the form $e^{i2\pi\theta}$.

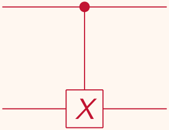


$$= \begin{bmatrix} e^{i2\pi\theta} & 0 \\ 0 & e^{i2\pi\theta} \end{bmatrix}$$

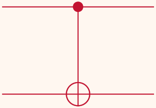


$$= \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{bmatrix}$$

- We can also make gates controlled:



$=$



$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Oracles

- Classical algorithms make calls to the memory to get the input.

Oracles

- Classical algorithms make calls to the memory to get the input.
- Quantum algorithms get an oracle that mimics this.

Oracles

- Classical algorithms make calls to the memory to get the input.
- Quantum algorithms get an oracle that mimics this.
- Oracle is a fancy word for black-box.

Oracles

- Classical algorithms make calls to the memory to get the input.
- Quantum algorithms get an oracle that mimics this.
- Oracle is a fancy word for black-box.
- A binary oracle for an input $x \in \{0,1\}^n$ is a unitary

$$O_x |i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle$$

where $\oplus$ is addition modulo 2 (or the XOR)

Oracles

- Classical algorithms make calls to the memory to get the input.
- Quantum algorithms get an <u>oracle</u> that mimics this.
- Oracle is a fancy word for black-box.
- A <u>binary oracle</u> for an input $x \in \{0,1\}^n$ is a unitary

$$O_x |i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle$$

  where $\oplus$ is addition modulo 2 (or the XOR)
- A <u>phase oracle</u> for an input $x \in \{0,1\}^n$ is a unitary

$$O_{x,\pm} |i\rangle = (-1)^{x_i} |i\rangle$$

Oracles

- Classical algorithms make calls to the memory to get the input.
- Quantum algorithms get an oracle that mimics this.
- Oracle is a fancy word for black-box.
- A binary oracle for an input $x \in \{0,1\}^n$ is a unitary

$$O_x |i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle$$

  where $\oplus$ is addition modulo 2 (or the XOR)
- A phase oracle for an input $x \in \{0,1\}^n$ is a unitary

$$O_{x,\pm} |i\rangle = (-1)^{x_i} |i\rangle$$

- We often want a controlled phase oracle:

$$O_{x,\pm} |i\rangle |0\rangle = |i\rangle |0\rangle , \qquad O_{x,\pm} |i\rangle |1\rangle = (-1)^{x_i} |i\rangle |1\rangle$$

Interpreting oracles

Let us look closer at a binary oracle.

$$O_x |i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle$$

Interpreting oracles

Let us look closer at a binary oracle.

$$O_x |i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle = \begin{cases} |i\rangle|b\rangle & \text{if } x_i = 0 \\ |i\rangle|\bar{b}\rangle & \text{if } x_i = 1 \end{cases}$$

Interpreting oracles

Let us look closer at a binary oracle.

$$O_x |i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle = \begin{cases} |i\rangle|b\rangle & \text{if } x_i = 0 \\ |i\rangle|\bar{b}\rangle & \text{if } x_i = 1 \end{cases}$$

So a X is applied to b if $x_i = 1$.

Interpreting oracles

Let us look closer at a binary oracle.

$$O_x |i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle = \begin{cases} |i\rangle|b\rangle \text{ if } x_i = 0 \\ |i\rangle|\bar{b}\rangle \text{ if } x_i = 1 \end{cases}$$

So a X is applied to b if $x_i = 1$.

Let us look closer at a controlled phase oracle.

$$O_{x,\pm} |i\rangle|b\rangle = \begin{cases} |i\rangle|b\rangle & \text{if } x_i = 0 \\ |i\rangle Z|b\rangle & \text{if } x_i = 1 \end{cases}$$

Interpreting oracles

Let us look closer at a binary oracle.

$$O_x |i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle = \begin{cases} |i\rangle|b\rangle \text{ if } x_i = 0 \\ |i\rangle|\bar{b}\rangle \text{ if } x_i = 1 \end{cases}$$

So a X is applied to b if $x_i = 1$.

Let us look closer at a controlled phase oracle.

$$O_{x,\pm} |i\rangle|b\rangle = \begin{cases} |i\rangle|b\rangle & \text{if } x_i = 0 \\ |i\rangle\, Z|b\rangle & \text{if } x_i = 1 \end{cases}$$

A Z is applied to b if $x_i = 1$.

Interpreting oracles

Let us look closer at a binary oracle.

$$O_x \, |i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle \; = \begin{cases} |i\rangle|b\rangle \ \text{ if } x_i = 0 \\ |i\rangle|\bar{b}\rangle \ \text{ if } x_i = 1 \end{cases}$$

So a X is applied to b if $x_i = 1$.

Let us look closer at a controlled phase oracle.

$$O_{x,\pm} \, |i\rangle|b\rangle = \begin{cases} |i\rangle|b\rangle & \text{if } x_i = 0 \\ |i\rangle \, Z|b\rangle & \text{if } x_i = 1 \end{cases}$$

A Z is applied to b if $x_i = 1$.

How would you convert between the two types of oracle?

Amplitude amplification

Randomized algorithms

Let us get back to classical computing for a while:

Randomized algorithms

Let us get back to classical computing for a while:

- An algorithm is randomized if it uses random bits.

Randomized algorithms

Let us get back to classical computing for a while:

- An algorithm is randomized if it uses random bits.
- Maybe our algorithm does not always work, it has a success probability p.

Randomized algorithms

Let us get back to classical computing for a while:

- An algorithm is randomized if it uses random bits.
- Maybe our algorithm does not always work, it has a success probability p.
- Formally, The algorithm outputs:

Randomized algorithms

Let us get back to classical computing for a while:

- An algorithm is randomized if it uses random bits.
- Maybe our algorithm does not always work, it has a success probability p.
- Formally, The algorithm outputs:
    - a 1 and a good solution with probability p, or

Randomized algorithms

Let us get back to classical computing for a while:

- An algorithm is randomized if it uses random bits.
- Maybe our algorithm does not always work, it has a success probability p.
- Formally, The algorithm outputs:
  - a 1 and a good solution with probability p, or
  - a 0 and a bad solution with probability $1 - p$.

Randomized algorithms

Let us get back to classical computing for a while:

- An algorithm is randomized if it uses random bits.
- Maybe our algorithm does not always work, it has a success probability p.
- Formally, The algorithm outputs:
  - a 1 and a good solution with probability p, or
  - a 0 and a bad solution with probability $1 - p$.
- An algorithm works with high probability if $p \geqslant 2/3$.

Randomized algorithms

Let us get back to classical computing for a while:

- An algorithm is randomized if it uses random bits.
- Maybe our algorithm does not always work, it has a success probability p.
- Formally, The algorithm outputs:
  - a 1 and a good solution with probability p, or
  - a 0 and a bad solution with probability $1 - p$.
- An algorithm works with high probability if $p \geqslant 2/3$.
- What can we do if p is small?

Randomized algorithms

Let us get back to classical computing for a while:

- An algorithm is randomized if it uses random bits.
- Maybe our algorithm does not always work, it has a success probability p.
- Formally, The algorithm outputs:
  - a 1 and a good solution with probability p, or
  - a 0 and a bad solution with probability $1 - p$.
- An algorithm works with high probability if $p \geqslant 2/3$.
- What can we do if p is small?
- Repeat

$$\mathcal{O}\left(\frac{1}{p}\right)$$

  times.

Example: prime finding

Consider to following algorithm:

Prime finding

1. Pick a random n-bit number k.
2. Check if k is prime.
3. Return whether k is prime and k itself.

Example: prime finding

Consider to following algorithm:

Prime finding

1. Pick a random n-bit number k.
2. Check if k is prime.
3. Return whether k is prime and k itself.

Probability of k being a prime is $\approx \frac{1}{n}$.

Example: prime finding

Consider to following algorithm:

Prime finding

1. Pick a random n-bit number k.
2. Check if k is prime.
3. Return whether k is prime and k itself.

Probability of k being a prime is $\approx \frac{1}{n}$.
$\Rightarrow \mathcal{O}(n)$ repetitions suffice.

Quantum algorithms with small success probability

Back to quantum:

Quantum algorithms with small success probability

Back to quantum:

- Quantum algorithms are inherently random.

Quantum algorithms with small success probability

Back to quantum:

- Quantum algorithms are inherently random.
- A Hadamard gate can be used to flip a coin:

$$H\,|0\rangle = \frac{1}{\sqrt{2}}\,(|0\rangle + |1\rangle)$$

Quantum algorithms with small success probability

Back to quantum:

- Quantum algorithms are inherently random.
- A Hadamard gate can be used to flip a coin:

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

- A quantum algorithm might produce the state

$$U|0\rangle = \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$$

Quantum algorithms with small success probability

Back to quantum:

- Quantum algorithms are inherently random.
- A Hadamard gate can be used to flip a coin:

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

- A quantum algorithm might produce the state

$$U|0\rangle = \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$$

- $|G\rangle |1\rangle$ is the "Good" part of the state, $|B\rangle |0\rangle$ is the "Bad" part.

Quantum algorithms with small success probability

Back to quantum:

- Quantum algorithms are inherently random.
- A Hadamard gate can be used to flip a coin:

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

- A quantum algorithm might produce the state

$$U|0\rangle = \alpha_G|G\rangle|1\rangle + \alpha_B|B\rangle|0\rangle$$

- $|G\rangle|1\rangle$ is the "Good" part of the state, $|B\rangle|0\rangle$ is the "Bad" part.
- Is we just measure then the success probability is $p = |\alpha_G|^2$.

Three states

Let us write $|\psi\rangle := \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$

Three states

Let us write $|\psi\rangle := \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$

- What is the inner product between the good and the bad part?

Three states

Let us write $|\psi\rangle := \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$

- What is the inner product between the good and the bad part?

$$\langle G|B\rangle\langle 1|0\rangle = \langle G|B\rangle \cdot 0 = 0$$

Three states

Let us write $|\psi\rangle := \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$

- What is the inner product between the good and the bad part?

$$\langle G|B\rangle\langle 1|0\rangle = \langle G|B\rangle \cdot 0 = 0$$

- So these two states are orthogonal!

Three states

Let us write $|\psi\rangle := \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$

- What is the inner product between the good and the bad part?

$$\langle G|B\rangle \langle 1|0\rangle = \langle G|B\rangle \cdot 0 = 0$$

- So these two states are orthogonal!
- $|\psi\rangle$ is written in the $\{|G\rangle|1\rangle, |B\rangle|0\rangle\}$ basis.

Three states

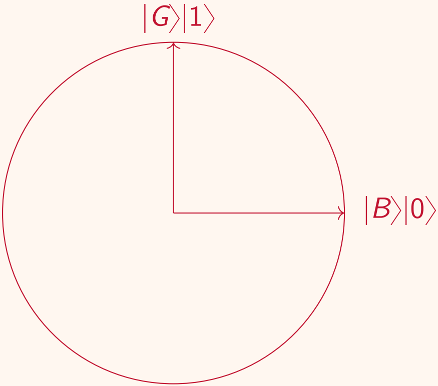
Let us write $|\psi\rangle := \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$

- What is the inner product between the good and the bad part?

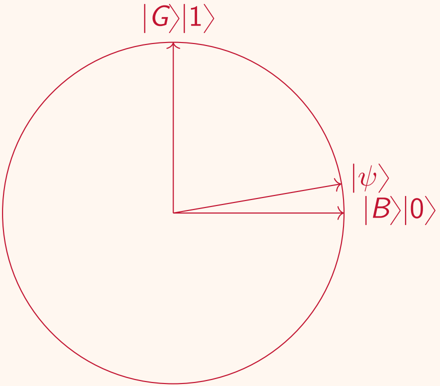
$$\langle G|B\rangle\langle 1|0\rangle = \langle G|B\rangle \cdot 0 = 0$$

- So these two states are orthogonal!
- $|\psi\rangle$ is written in the $\{|G\rangle|1\rangle, |B\rangle|0\rangle\}$ basis.
- Everything is in a 2-dimensional subspace.

Three states - a picture

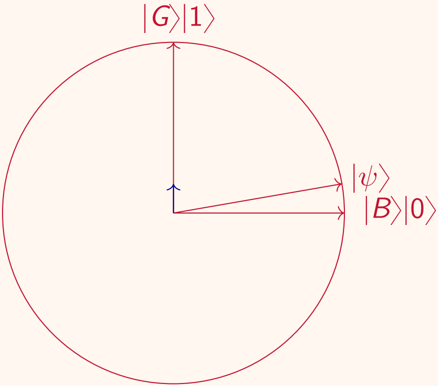


Three states - a picture



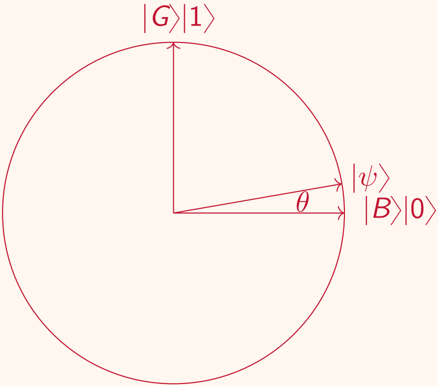
$$|\psi\rangle = \alpha_G |G\rangle |1\rangle + \alpha_B |B\rangle |0\rangle$$

Three states - a picture



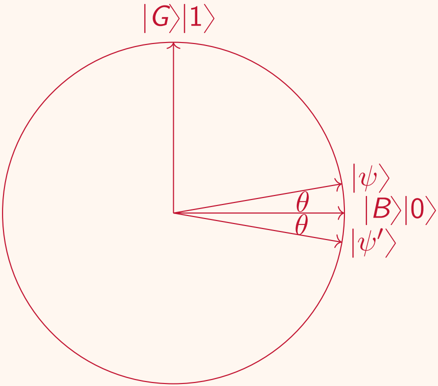
$$|\psi\rangle = \alpha_G\,|G\rangle\,|1\rangle + \alpha_B\,|B\rangle\,|0\rangle$$

Three states - a picture



$$|\psi\rangle = \sin(\theta)\ |G\rangle|1\rangle + \cos(\theta)\ |B\rangle|0\rangle$$

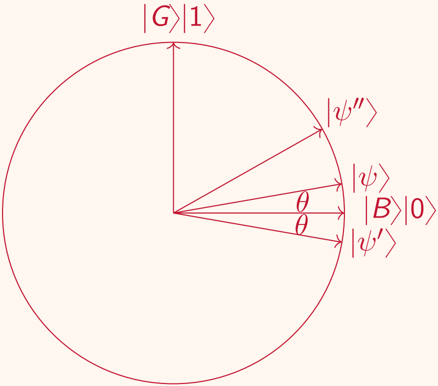
Three states - a picture



$$|\psi\rangle = \sin(\theta)\; |G\rangle|1\rangle + \cos(\theta)\; |B\rangle|0\rangle$$

$$|\psi'\rangle = -\sin(\theta)\,|G\rangle|1\rangle + \cos(\theta)\,|B\rangle|0\rangle$$

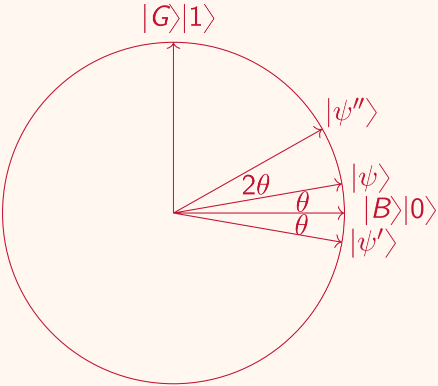
Three states - a picture



$$|\psi\rangle = \sin(\theta) \; |G\rangle|1\rangle + \cos(\theta) \; |B\rangle|0\rangle$$

$$|\psi'\rangle = -\sin(\theta)\,|G\rangle|1\rangle + \cos(\theta)\,|B\rangle|0\rangle$$

Three states - a picture

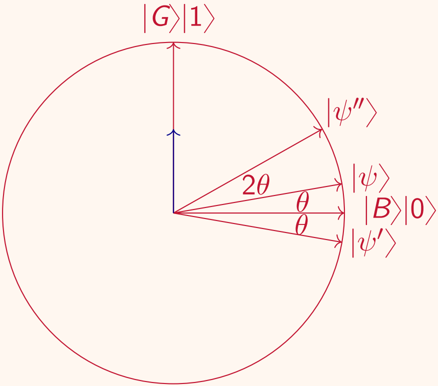


$$|\psi\rangle = \sin(\theta)\ |G\rangle|1\rangle + \cos(\theta)\ |B\rangle|0\rangle$$

$$|\psi'\rangle = -\sin(\theta)\,|G\rangle|1\rangle + \cos(\theta)\,|B\rangle|0\rangle$$

$$|\psi''\rangle = \sin(3\theta)\,|G\rangle|1\rangle + \cos(3\theta)\,|B\rangle|0\rangle$$

Three states - a picture



$$|\psi\rangle = \sin(\theta)\ |G\rangle|1\rangle + \cos(\theta)\ |B\rangle|0\rangle$$

$$|\psi'\rangle = -\sin(\theta)|G\rangle|1\rangle + \cos(\theta)|B\rangle|0\rangle$$

$$|\psi''\rangle = \sin(3\theta)|G\rangle|1\rangle + \cos(3\theta)|B\rangle|0\rangle$$

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a rotation $\mathcal{A}$, with angle 2θ.

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a <u>rotation</u> $\mathcal{A}$, with angle 2θ.
- After k iterations of $\mathcal{A}$ we get

$$\sin\left((2k+1)\theta\right)|G\rangle|1\rangle + \cos\left((2k+1)\right)|B\rangle|0\rangle$$

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a rotation $\mathcal{A}$, with angle 2θ.
- After k iterations of $\mathcal{A}$ we get

$$\sin\left((2k+1)\theta\right)|G\rangle|1\rangle + \cos\left((2k+1)\right)|B\rangle|0\rangle$$

- If $(2k+1)\theta \approx \pi/2$ then $\left|\sin\left((2k+1)\theta\right)\right|^2 \approx 1$.

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a rotation $\mathcal{A}$, with angle 2θ.
- After k iterations of $\mathcal{A}$ we get

$$\sin\left((2k+1)\theta\right)|G\rangle|1\rangle + \cos\left((2k+1)\right)|B\rangle|0\rangle$$

- If $(2k+1)\theta \approx \pi/2$ then $\left|\sin\left((2k+1)\theta\right)\right|^2 \approx 1$.
- Since $\theta = \arcsin \alpha_G$ we want

$$k \approx \left(\frac{\pi}{2\arcsin \alpha_G} - 1\right)/2$$

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a <u>rotation</u> $\mathcal{A}$, with angle 2θ.
- After k iterations of $\mathcal{A}$ we get

$$\sin\left((2k+1)\theta\right)|G\rangle|1\rangle + \cos\left((2k+1)\right)|B\rangle|0\rangle$$

- If $(2k+1)\theta \approx \pi/2$ then $\left|\sin\left((2k+1)\theta\right)\right|^2 \approx 1$.
- Since $\theta = \arcsin\alpha_G$ we want

$$k \approx \left(\frac{\pi}{2\arcsin\alpha_G} - 1\right)/2 = \mathcal{O}\left(\frac{1}{|\alpha_G|}\right)$$

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a <u>rotation</u> $\mathcal{A}$, with angle 2θ.
- After k iterations of $\mathcal{A}$ we get

$$\sin\left((2k+1)\theta\right)|G\rangle|1\rangle + \cos\left((2k+1)\right)|B\rangle|0\rangle$$

- If $(2k+1)\theta \approx \pi/2$ then $\left|\sin\left((2k+1)\theta\right)\right|^2 \approx 1$.
- Since $\theta = \arcsin\alpha_G$ we want

$$k \approx \left(\frac{\pi}{2\arcsin\alpha_G} - 1\right)/2 = \mathcal{O}\left(\frac{1}{|\alpha_G|}\right) = \mathcal{O}\left(\frac{1}{\sqrt{p}}\right)$$

Amplitude amplification

- Two reflections: through $|B\rangle|0\rangle$ and $|\psi\rangle$.
- The product is a rotation $\mathcal{A}$, with angle 2θ.
- After k iterations of $\mathcal{A}$ we get

$$\sin\left((2k+1)\theta\right)|G\rangle|1\rangle + \cos\left((2k+1)\right)|B\rangle|0\rangle$$

- If $(2k+1)\theta \approx \pi/2$ then $|\sin\left((2k+1)\theta\right)|^2 \approx 1$.
- Since $\theta = \arcsin \alpha_G$ we want

$$k \approx \left(\frac{\pi}{2\arcsin \alpha_G} - 1\right)/2 = \mathcal{O}\left(\frac{1}{|\alpha_G|}\right) = \mathcal{O}\left(\frac{1}{\sqrt{p}}\right)$$

- Nice, but can we actually implement these reflections?

Implementing the reflections

The reflection through $|B\rangle|0\rangle$:

Implementing the reflections

The reflection through $|B\rangle|0\rangle$:

- Do nothing to the bad state.
- Add a -1 to the good state.

Implementing the reflections

The reflection through $|B\rangle|0\rangle$:

- Do nothing to the bad state.
- Add a -1 to the good state.

Apply a Z gate to the last bit.

Implementing the reflections

The reflection through $|B\rangle|0\rangle$:

- Do nothing to the bad state.
- Add a -1 to the good state.

Apply a Z gate to the last bit.

The reflection through $|\psi\rangle$:

Implementing the reflections

The reflection through $|B\rangle|0\rangle$:

- Do nothing to the bad state.
- Add a -1 to the good state.

Apply a Z gate to the last bit.

The reflection through $|\psi\rangle$:

- Do nothing to $|\psi\rangle$.
- Add a -1 to states orthogonal to it.

Implementing the reflections

The reflection through $|B\rangle|0\rangle$:

- Do nothing to the bad state.
- Add a -1 to the good state.

Apply a Z gate to the last bit.

The reflection through $|\psi\rangle$:

- Do nothing to $|\psi\rangle$.
- Add a -1 to states orthogonal to it.

Use that $|\psi\rangle = U|0\rangle$:

1. Apply U^{-1} to map $|\psi\rangle$ to $|0\rangle$.
2. Reflect through $|0\rangle$.
3. Apply U to map $|0\rangle$ to back to $|\psi\rangle$.

Example: the search problem

Before amplitude amplification there was Grover ('96).

Example: the search problem

Before amplitude amplification there was Grover ('96).

Search problem

Input: $x \in \{0,1\}^n$ with k ones.
Goal: Find an i such that $x_i = 1$ with few queries.

Example: the search problem

Before amplitude amplification there was Grover ('96).

Search problem

Input: $x \in \{0,1\}^n$ with k ones.
Goal: Find an i such that $x_i = 1$ with few queries.

How can we do this classically?

Example: the search problem

Before amplitude amplification there was Grover ('96).

Search problem

Input: $x \in \{0,1\}^n$ with k ones.
Goal: Find an i such that $x_i = 1$ with few queries.

How can we do this classically?

- Go over the bits: $\mathcal{O}(n)$ queries.

Example: the search problem

Before amplitude amplification there was Grover ('96).

Search problem

Input: $x \in \{0,1\}^n$ with k ones.
Goal: Find an i such that $x_i = 1$ with few queries.

How can we do this classically?

- Go over the bits: $\mathcal{O}(n)$ queries.
- Randomly pick an i and repeat: $\mathcal{O}(n/k)$ queries.

Example: the search problem

And with a quantum algorithm?

Example: the search problem

And with a quantum algorithm?
Let U be:

- Setup a uniform superposition over the i by applying H to all bits.

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |0\rangle$$

Example: the search problem

And with a quantum algorithm?
Let U be:

- Setup a uniform superposition over the i by applying H to all bits.

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |0\rangle$$

- Query in superposition

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |x_i\rangle$$

Example: the search problem

And with a quantum algorithm?
Let U be:

- Setup a uniform superposition over the i by applying H to all bits.

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |0\rangle$$

- Query in superposition

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |x_i\rangle = \left(\quad \sum_{i:x_i=1} |i\rangle \right) |1\rangle$$

Example: the search problem

And with a quantum algorithm?
Let U be:

- Setup a uniform superposition over the i by applying H to all bits.

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |0\rangle$$

- Query in superposition

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |x_i\rangle = \qquad \left(\frac{1}{\sqrt{k}} \sum_{i:x_i=1} |i\rangle \right) |1\rangle$$

Example: the search problem

And with a quantum algorithm?
Let U be:

- Setup a uniform superposition over the i by applying H to all bits.

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |0\rangle$$

- Query in superposition

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |x_i\rangle = \sqrt{\frac{k}{n}} \left(\frac{1}{\sqrt{k}} \sum_{i:x_i=1} |i\rangle \right) |1\rangle$$

Example: the search problem

And with a quantum algorithm?
Let U be:

- Setup a uniform superposition over the i by applying H to all bits.

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |0\rangle$$

- Query in superposition

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |x_i\rangle = \sqrt{\frac{k}{n}} \left(\frac{1}{\sqrt{k}} \sum_{i:x_i=1} |i\rangle \right) |1\rangle + (\dots) |0\rangle$$

Example: the search problem

And with a quantum algorithm?
Let U be:

- Setup a uniform superposition over the i by applying H to all bits.

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |0\rangle$$

- Query in superposition

$$\frac{1}{\sqrt{n}} \sum_{i=0}^{n} |i\rangle |x_i\rangle = \sqrt{\frac{k}{n}} \left(\frac{1}{\sqrt{k}} \sum_{i:x_i=1} |i\rangle \right) |1\rangle + (\dots) |0\rangle$$

With $\mathcal{O}\left(\sqrt{\frac{n}{k}}\right)$ iterations we are done!

Phase estimation

Eigenvalues of unitaries

Eigenvectors and eigenvalues

If

$$U|\psi\rangle = \lambda\,|\psi\rangle$$

then $|\psi\rangle$ is an eigenvector and λ is the corresponding eigenvalue.

Eigenvalues of unitaries

Eigenvectors and eigenvalues

If

$$U|\psi\rangle = \lambda\,|\psi\rangle$$

then $|\psi\rangle$ is an eigenvector and λ is the corresponding eigenvalue.

For example

$$Z|0\rangle = \quad 1 \cdot |0\rangle$$

Eigenvalues of unitaries

Eigenvectors and eigenvalues

If

$$U|\psi\rangle = \lambda\,|\psi\rangle$$

then $|\psi\rangle$ is an eigenvector and λ is the corresponding eigenvalue.

For example

$$\begin{aligned} Z|0\rangle &= 1\cdot|0\rangle \\ Z|1\rangle &= -1\cdot|1\rangle \end{aligned}$$

Eigenvalues of unitaries

Eigenvectors and eigenvalues

If

$$U|\psi\rangle = \lambda|\psi\rangle$$

then $|\psi\rangle$ is an eigenvector and λ is the corresponding eigenvalue.

For example

$$Z|0\rangle = \quad 1 \cdot |0\rangle$$

$$Z|1\rangle = -1 \cdot |1\rangle$$

In general for a unitary U and eigenvector $|\psi\rangle$ of U:

$$U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$$

Remember: $e^{i\theta}$ is called a phase.

Goal of phase estimation

Goal

Given a U and $|\psi\rangle$, s.t.,

$$U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$$

calculate θ (up to some precision) by applying U in some way.

Goal of phase estimation

Goal

Given a U and $|\psi\rangle$, s.t.,

$$U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$$

calculate θ (up to some precision) by applying U in some way.

Example: if we know that $U = I$ or $U = Z$ and do phase estimation on $|\psi\rangle = |1\rangle$:

Goal of phase estimation

Goal

Given a U and $|\psi\rangle$, s.t.,

$$U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$$

calculate θ (up to some precision) by applying U in some way.

Example: if we know that $U = I$ or $U = Z$ and do phase estimation on $|\psi\rangle = |1\rangle$:

- If $U = I$ then $U|1\rangle = 1 \cdot |1\rangle = e^{i2\pi 0}|1\rangle$ so $\theta = 0$.

Goal of phase estimation

Goal

Given a U and $|\psi\rangle$, s.t.,

$$U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$$

calculate θ (up to some precision) by applying U in some way.

Example: if we know that $U = I$ or $U = Z$ and do phase estimation on $|\psi\rangle = |1\rangle$:

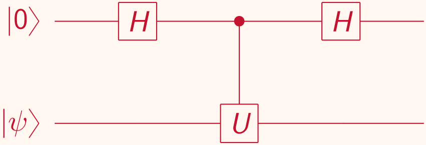
- If $U = I$ then $U|1\rangle = 1 \cdot |1\rangle = e^{i2\pi 0}|1\rangle$ so $\theta = 0$.
- If $U = Z$ then $U|1\rangle = -1 \cdot |1\rangle = e^{i2\pi\frac{1}{2}}|1\rangle$ so $\theta = \frac{1}{2}$.

A $+1$ or -1 eigenvalue

What can we do if U either applies a $+1$ or -1 phase and we want a binary description of this phase?

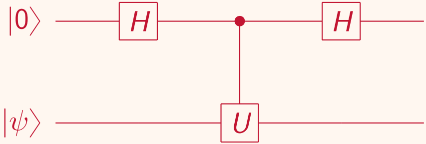
A $+1$ or -1 eigenvalue

What can we do if U either applies a $+1$ or -1 phase and we want a binary description of this phase? We use our oracle conversion trick!



A $+1$ or -1 eigenvalue

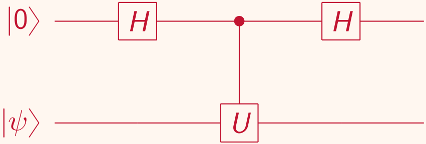
What can we do if U either applies a $+1$ or -1 phase and we want a binary description of this phase? We use our oracle conversion trick!



Let's check this for a $-1 = e^{i2\pi\frac{1}{2}}$ phase:

A $+1$ or -1 eigenvalue

What can we do if U either applies a $+1$ or -1 phase and we want a binary description of this phase? We use our oracle conversion trick!

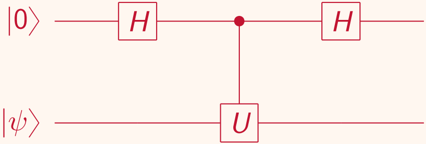


Let's check this for a $-1 = e^{i2\pi\frac{1}{2}}$ phase:

- We start in $|0\rangle|\psi\rangle$.

A $+1$ or -1 eigenvalue

What can we do if U either applies a $+1$ or -1 phase and we want a binary description of this phase? We use our oracle conversion trick!

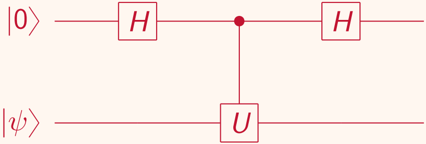


Let's check this for a $-1 = e^{i2\pi\frac{1}{2}}$ phase:

- We start in $|0\rangle|\psi\rangle$.
- We apply H to get $\frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle + |1\rangle|\psi\rangle)$.

A $+1$ or -1 eigenvalue

What can we do if U either applies a $+1$ or -1 phase and we want a binary description of this phase? We use our oracle conversion trick!

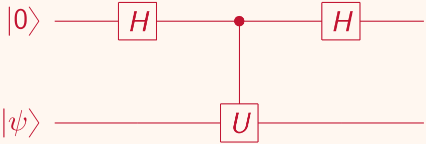


Let's check this for a $-1 = e^{i2\pi\frac{1}{2}}$ phase:

- We start in $|0\rangle|\psi\rangle$.
- We apply H to get $\frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle + |1\rangle|\psi\rangle)$.
- We apply U only to the $|1\rangle$ part:
  $\frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle + |1\rangle\, U\,|\psi\rangle) = \frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle - |1\rangle\,|\psi\rangle)$.

A $+1$ or -1 eigenvalue

What can we do if U either applies a $+1$ or -1 phase and we want a binary description of this phase? We use our oracle conversion trick!



Let's check this for a $-1 = e^{i2\pi\frac{1}{2}}$ phase:

- We start in $|0\rangle|\psi\rangle$.
- We apply H to get $\frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle + |1\rangle|\psi\rangle)$.
- We apply U only to the $|1\rangle$ part:
  $\frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle + |1\rangle\, U\,|\psi\rangle) = \frac{1}{\sqrt{2}}(|0\rangle|\psi\rangle - |1\rangle\,|\psi\rangle)$.
- We apply H again and get $|1\rangle|\psi\rangle$.

A bit about binary

What if we want to estimate other phases as well?

A bit about binary

What if we want to estimate other phases as well?

- In general a phase is $e^{i2\pi\theta}$.

A bit about binary

What if we want to estimate other phases as well?

- In general a phase is $e^{i2\pi\theta}$.
- We can write θ in binary:

$$\theta = 10.110101$$

A bit about binary

What if we want to estimate other phases as well?

- In general a phase is $e^{i2\pi\theta}$.
- We can write θ in binary:

$$\theta = 10.110101$$

- The part before the decimal does not matter since

$$e^{i2\pi 1} = e^{i2\pi 2} = e^{i2\pi 3} = \dots$$

A bit about binary

What if we want to estimate other phases as well?

- In general a phase is $e^{i2\pi\theta}$.
- We can write θ in binary:

$$\theta = 10.110101$$

- The part before the decimal does not matter since

$$e^{i2\pi 1} = e^{i2\pi 2} = e^{i2\pi 3} = \dots$$

- So we only need to write down the part behind the decimal.

A bit about binary

What if we want to estimate other phases as well?

- In general a phase is $e^{i2\pi\theta}$.
- We can write θ in binary:

$$\theta = 10.110101$$

- The part before the decimal does not matter since

$$e^{i2\pi 1} = e^{i2\pi 2} = e^{i2\pi 3} = \dots$$

- So we only need to write down the part behind the decimal.
- A phase of $1 = e^{i2\pi 0.0}$ and a phase of $-1 = e^{i2\pi \frac{1}{2}} = e^{i2\pi 0.1}$ (in binary!)

A bit about binary

What if we want to estimate other phases as well?

- In general a phase is $e^{i2\pi\theta}$.
- We can write θ in binary:

$$\theta = 10.110101$$

- The part before the decimal does not matter since

$$e^{i2\pi 1} = e^{i2\pi 2} = e^{i2\pi 3} = \dots$$

- So we only need to write down the part behind the decimal.
- A phase of $1 = e^{i2\pi 0.0}$ and a phase of $-1 = e^{i2\pi \frac{1}{2}} = e^{i2\pi 0.1}$ (in binary!)
- If this is only one bit ($\theta = 0.0$ or $\theta = 0.1 = \frac{1}{2}$) then we know what to do.

Last bit phase estimation

Let $U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$ for $\theta = 0.b_3b_2b_1$, then we can calculate b_1!

Last bit phase estimation

Let $U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$ for $\theta = 0.b_3b_2b_1$, then we can calculate b_1!

- We look at U^4, so $U^4|\psi\rangle = \left(e^{i2\pi\theta}\right)^4|\psi\rangle = e^{i2\pi 4\theta}|\psi\rangle$.

Last bit phase estimation

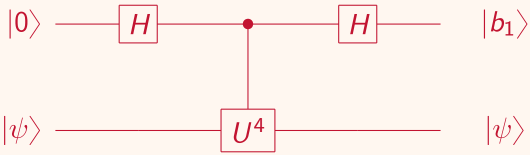
Let $U\left|\psi\right\rangle = e^{i2\pi\theta}\left|\psi\right\rangle$ for $\theta = 0.b_3b_2b_1$, then we can calculate b_1!

- We look at U^4, so $U^4\left|\psi\right\rangle = \left(e^{i2\pi\theta}\right)^4\left|\psi\right\rangle = e^{i2\pi 4\theta}\left|\psi\right\rangle$.
- Now the phase corresponds with $4\theta = b_3b_2.b_1$

Last bit phase estimation

Let $U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$ for $\theta = 0.b_3b_2b_1$, then we can calculate b_1!

- We look at U^4, so $U^4|\psi\rangle = \left(e^{i2\pi\theta}\right)^4|\psi\rangle = e^{i2\pi 4\theta}|\psi\rangle$.
- Now the phase corresponds with $4\theta = b_3b_2.b_1$
- Simply use our previous circuit to get the last bit:



Two bit phase estimation

Let $U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$ for $\theta = 0.b_3b_2b_1$, can we calculate b_2 as well?

Two bit phase estimation

Let $U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$ for $\theta = 0.b_3b_2b_1$, can we calculate b_2 as well?

- We could look at U^2? The phase corresponds with $2\theta = b_3.b_2b_1$.

Two bit phase estimation

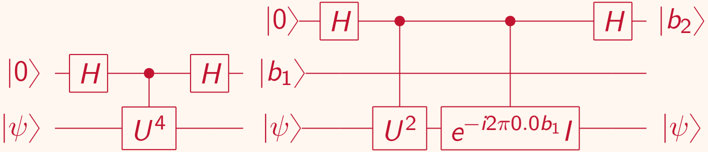
Let $U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$ for $\theta = 0.b_3b_2b_1$, can we calculate b_2 as well?

- We could look at U^2? The phase corresponds with $2\theta = b_3.b_2b_1$.
- If $b_1 = 0$ then we can just use our algorithm. But what if $b_1 = 1$?

Two bit phase estimation

Let $U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$ for $\theta = 0.b_3b_2b_1$, can we calculate b_2 as well?

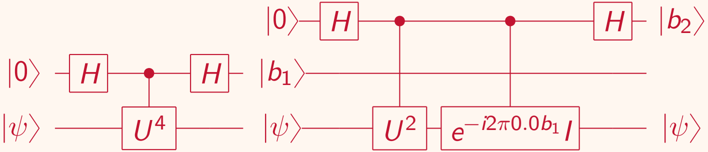
- We could look at U^2? The phase corresponds with $2\theta = b_3.b_2b_1$.
- If $b_1 = 0$ then we can just use our algorithm. But what if $b_1 = 1$?
- Solution: calculate b_1 and then make a change to U if necessary:



Two bit phase estimation

Let $U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$ for $\theta = 0.b_3b_2b_1$, can we calculate b_2 as well?

- We could look at U^2? The phase corresponds with $2\theta = b_3.b_2b_1$.
- If $b_1 = 0$ then we can just use our algorithm. But what if $b_1 = 1$?
- Solution: calculate b_1 and then make a change to U if necessary:



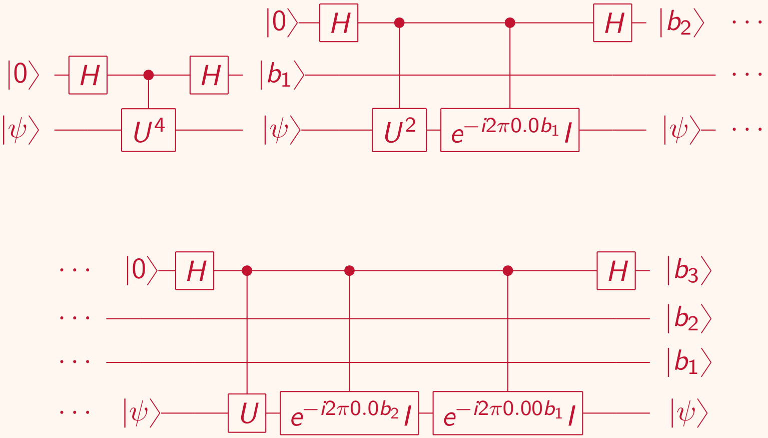
- This works because $U^2 e^{-i2\pi 0.b_1}$ adds a phase with $0.b_2b_1 - 0.0b_1 = 0.b_2$.

Three bit phase estimation

Let $U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$ for $\theta = 0.b_3b_2b_1$, can we calculate all the bits?

Three bit phase estimation

Let $U|\psi\rangle = e^{i2\pi\theta}|\psi\rangle$ for $\theta = 0.b_3b_2b_1$, can we calculate all the bits? Yes! just keep going:

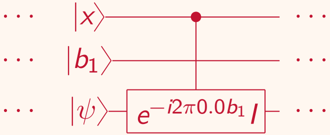


Clean up

Can we clean up this circuit?

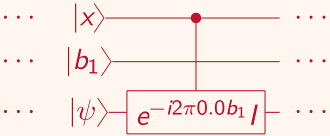
Clean up

Can we clean up this circuit? Consider



Clean up

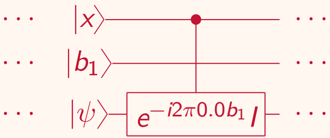
Can we clean up this circuit? Consider



This applies a phase of $e^{-i2\pi 0.0b_1}$ to the state if $x = 1$.

Clean up

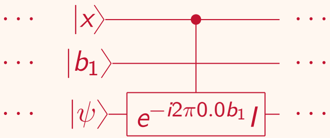
Can we clean up this circuit? Consider



This applies a phase of $e^{-i2\pi 0.0b_1}$ to the state if $x = 1$.
$\Leftrightarrow$ Applies a phase of $e^{-i\pi/2}$ to the state is $x = 1$ and $b_1 = 1$.

Clean up

Can we clean up this circuit? Consider



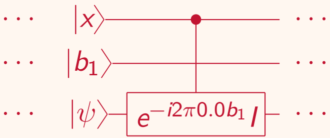
This applies a phase of $e^{-i2\pi 0.0b_1}$ to the state if $x = 1$.

$\Leftrightarrow$ Applies a phase of $e^{-i\pi/2}$ to the state is $x = 1$ and $b_1 = 1$.

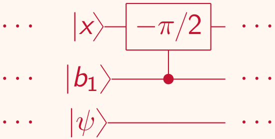
$\Leftrightarrow$ Applies a phase gate $-\boxed{-\pi/2}-$ to $|x\rangle$ if $b_1 = 1$.

Clean up

Can we clean up this circuit? Consider

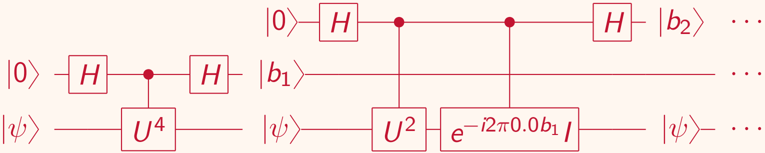


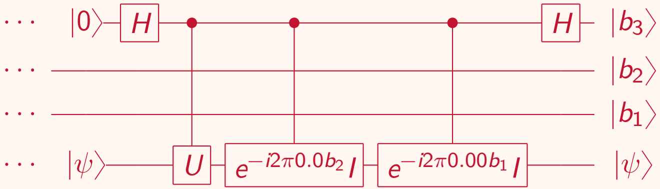
This applies a phase of $e^{-i2\pi 0.0b_1}$ to the state if $x = 1$.
$\Leftrightarrow$ Applies a phase of $e^{-i\pi/2}$ to the state is $x = 1$ and $b_1 = 1$.
$\Leftrightarrow$ Applies a phase gate $-\boxed{-\pi/2}-$ to $|x\rangle$ if $b_1 = 1$.



The inverse quantum Fourier transform

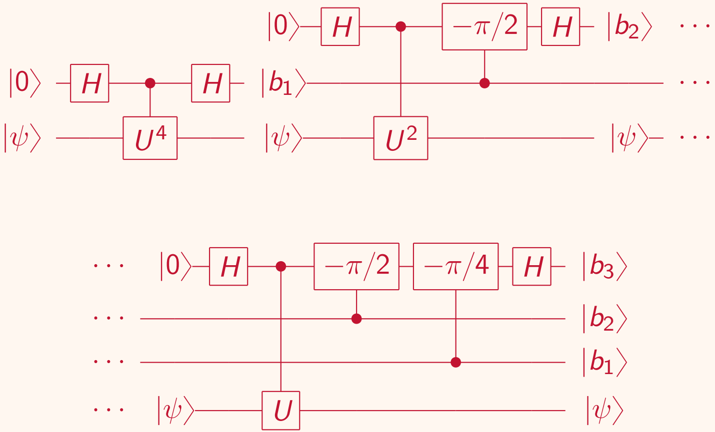
Can we clean up this circuit?





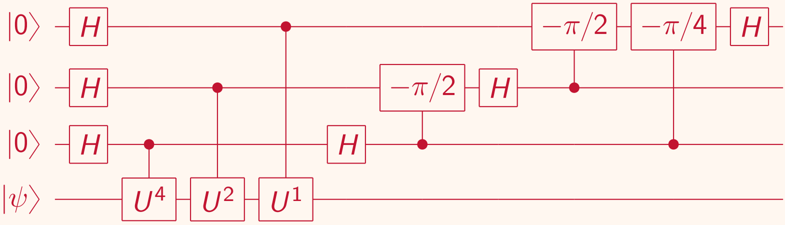
The inverse quantum Fourier transform

Can we clean up this circuit?



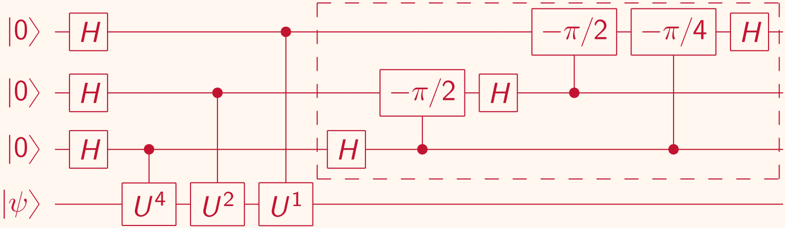
The inverse quantum Fourier transform

Can we clean up this circuit?



The inverse quantum Fourier transform

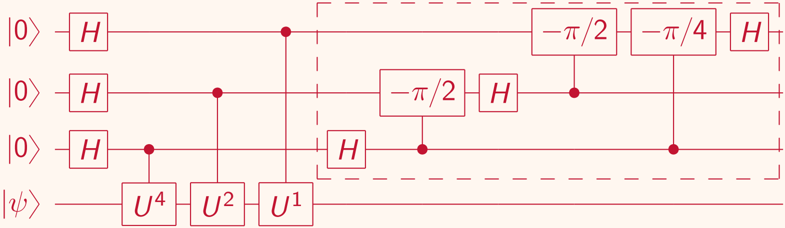
Can we clean up this circuit?



This circuit is called the inverse quantum Fourier transform (QFT^{-1}).

The inverse quantum Fourier transform

Can we clean up this circuit?

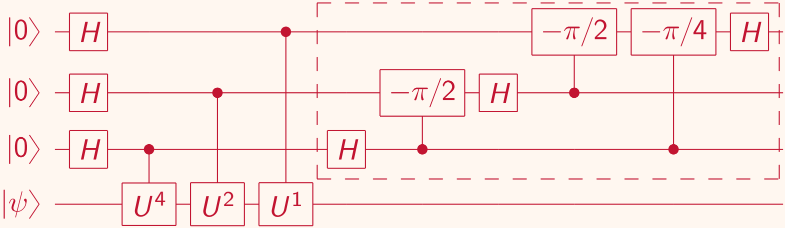


This circuit is called the <u>inverse quantum Fourier transform</u> (QFT^{-1}).

(The inverse quantum Fourier transform actually also swaps the order of the qubits, but this does not matter for our purpose)

The inverse quantum Fourier transform

Can we clean up this circuit?



This circuit is called the inverse quantum Fourier transform (QFT^{-1}).

(The inverse quantum Fourier transform actually also swaps the order of the qubits, but this does not matter for our purpose)

In total we need to apply U^k for $k \sim \frac{1}{\varepsilon}$

Example: finding an eigenvector

What happens if $|\psi\rangle$ is not an eigenvector?

Example: finding an eigenvector

What happens if $|\psi\rangle$ is not an eigenvector?

- Let
$$|\psi\rangle = \alpha_0 |\phi_0\rangle + \alpha_1 |\phi_1\rangle$$
where $U|\phi_0\rangle = e^{i2\pi\theta_0} |\phi\rangle$ and $U|\phi_1\rangle = e^{i2\pi\theta_1} |\phi\rangle$.

Example: finding an eigenvector

What happens if $|\psi\rangle$ is not an eigenvector?

- Let
$$|\psi\rangle = \alpha_0 |\phi_0\rangle + \alpha_1 |\phi_1\rangle$$
where $U|\phi_0\rangle = e^{i2\pi\theta_0} |\phi\rangle$ and $U|\phi_1\rangle = e^{i2\pi\theta_1} |\phi\rangle$.
- If we apply phase estimation we get (by linearity):
$$PE_U |0\rangle|\psi\rangle = \alpha_0 |\theta_0\rangle|\phi_0\rangle + \alpha_1 |\theta_1\rangle|\phi_1\rangle$$

Example: finding an eigenvector

What happens if $|\psi\rangle$ is not an eigenvector?

- Let
$$|\psi\rangle = \alpha_0 |\phi_0\rangle + \alpha_1 |\phi_1\rangle$$
where $U|\phi_0\rangle = e^{i2\pi\theta_0} |\phi\rangle$ and $U|\phi_1\rangle = e^{i2\pi\theta_1} |\phi\rangle$.
- If we apply phase estimation we get (by linearity):
$$PE_U |0\rangle|\psi\rangle = \alpha_0 |\theta_0\rangle|\phi_0\rangle + \alpha_1 |\theta_1\rangle|\phi_1\rangle$$
- Measuring the first register gives us a random θ_i, what happens to the second register?

Amplitude estimation

A problem with amplitude amplification

In amplitude amplification we had the state

$$|\psi\rangle = \alpha_G \, |G\rangle|1\rangle + \alpha_B \, |B\rangle|0\rangle$$

A problem with amplitude amplification

In amplitude amplification we had the state

$$|\psi\rangle = \alpha_G |G\rangle|1\rangle + \alpha_B |B\rangle|0\rangle = \sin(\theta)\,|G\rangle|1\rangle + \cos(\theta)\,|B\rangle|0\rangle\,.$$

A problem with amplitude amplification

In amplitude amplification we had the state

$$|\psi\rangle = \alpha_G |G\rangle|1\rangle + \alpha_B |B\rangle|0\rangle = \sin(\theta)|G\rangle|1\rangle + \cos(\theta)|B\rangle|0\rangle.$$

- Each iteration applies a rotation by 2θ, increasing the angle.

A problem with amplitude amplification

In amplitude amplification we had the state

$$|\psi\rangle = \alpha_G |G\rangle|1\rangle + \alpha_B |B\rangle|0\rangle = \sin(\theta) |G\rangle|1\rangle + \cos(\theta) |B\rangle|0\rangle .$$

- Each iteration applies a rotation by 2θ, increasing the angle.
- But how many iterations are needed? We might not know θ.

A problem with amplitude amplification

In amplitude amplification we had the state

$$|\psi\rangle = \alpha_G |G\rangle|1\rangle + \alpha_B |B\rangle|0\rangle = \sin(\theta)|G\rangle|1\rangle + \cos(\theta)|B\rangle|0\rangle .$$

- Each iteration applies a rotation by 2θ, increasing the angle.
- But how many iterations are needed? We might not know θ.
- For example for search: how many solutions are there?

A problem with amplitude amplification

In amplitude amplification we had the state

$$|\psi\rangle = \alpha_G |G\rangle|1\rangle + \alpha_B |B\rangle|0\rangle = \sin(\theta) |G\rangle|1\rangle + \cos(\theta) |B\rangle|0\rangle .$$

- Each iteration applies a rotation by 2θ, increasing the angle.
- But how many iterations are needed? We might not know θ.
- For example for search: how many solutions are there?
- Too many iterations decreases the success probability again!

A closer look at the amplitude amplification iterate

In the basis $\{|G\rangle|1\rangle, |B\rangle|0\rangle\}$ the iterate is a rotation:

$$\begin{bmatrix} \cos(2\theta) & -\sin(2\theta) \\ \sin(2\theta) & \cos(2\theta) \end{bmatrix}$$

A closer look at the amplitude amplification iterate

In the basis $\{|G\rangle|1\rangle, |B\rangle|0\rangle\}$ the iterate is a rotation:

$$\begin{bmatrix} \cos(2\theta) & -\sin(2\theta) \\ \sin(2\theta) & \cos(2\theta) \end{bmatrix}$$

What are the eigenvalues of a rotation?

A closer look at the amplitude amplification iterate

In the basis $\{|G\rangle|1\rangle, |B\rangle|0\rangle\}$ the iterate is a rotation:

$$\begin{bmatrix} \cos(2\theta) & -\sin(2\theta) \\ \sin(2\theta) & \cos(2\theta) \end{bmatrix}$$

What are the eigenvalues of a rotation?

$$e^{i2\theta}, e^{-i2\theta}$$

A closer look at the amplitude amplification iterate

In the basis $\{|G\rangle|1\rangle, |B\rangle|0\rangle\}$ the iterate is a rotation:

$$\begin{bmatrix} \cos(2\theta) & -\sin(2\theta) \\ \sin(2\theta) & \cos(2\theta) \end{bmatrix}$$

What are the eigenvalues of a rotation?

$$e^{i2\theta}, e^{-i2\theta}$$

The eigenvectors also lie in the 2-dimensional subspace.

Amplitude estimation

Amplitude estimation

Input: A unitary that prepares

$$|\psi\rangle = \alpha_G \, |G\rangle|1\rangle + \alpha_B \, |B\rangle|0\rangle \, .$$

Goal: Estimate the probability of $|G\rangle|1\rangle$ up to error ε.

Amplitude estimation

Amplitude estimation

Input: A unitary that prepares

$$|\psi\rangle = \alpha_G |G\rangle|1\rangle + \alpha_B |B\rangle|0\rangle .$$

Goal: Estimate the probability of $|G\rangle|1\rangle$ up to error ε.

We use phase estimation on the amplitude amplification iterate!

Amplitude estimation

Amplitude estimation

Input: A unitary that prepares

$$|\psi\rangle = \alpha_G |G\rangle|1\rangle + \alpha_B |B\rangle|0\rangle .$$

Goal: Estimate the probability of $|G\rangle|1\rangle$ up to error ε.

We use phase estimation on the amplitude amplification iterate!

- The eigenvectors form a (orthonormal) basis for the two dimensional space.

Amplitude estimation

Amplitude estimation

Input: A unitary that prepares

$$|\psi\rangle = \alpha_G |G\rangle|1\rangle + \alpha_B |B\rangle|0\rangle .$$

Goal: Estimate the probability of $|G\rangle|1\rangle$ up to error ε.

We use phase estimation on the amplitude amplification iterate!

- The eigenvectors form a (orthonormal) basis for the two dimensional space.
- $|\psi\rangle$ is some linear combination of the eigenvectors.

Amplitude estimation

Amplitude estimation

Input: A unitary that prepares

$$|\psi\rangle = \alpha_G |G\rangle|1\rangle + \alpha_B |B\rangle|0\rangle .$$

Goal: Estimate the probability of $|G\rangle|1\rangle$ up to error ε.

We use phase estimation on the amplitude amplification iterate!

- The eigenvectors form a (orthonormal) basis for the two dimensional space.
- $|\psi\rangle$ is some linear combination of the eigenvectors.
- Using phase estimation we find either 2θ or -2θ.

Amplitude estimation

Amplitude estimation

Input: A unitary that prepares

$$|\psi\rangle = \alpha_G |G\rangle|1\rangle + \alpha_B |B\rangle|0\rangle .$$

Goal: Estimate the probability of $|G\rangle|1\rangle$ up to error ε.

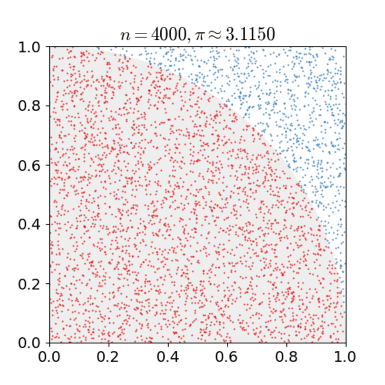
We use phase estimation on the amplitude amplification iterate!

- The eigenvectors form a (orthonormal) basis for the two dimensional space.
- $|\psi\rangle$ is some linear combination of the eigenvectors.
- Using phase estimation we find either 2θ or -2θ.
- Both tell us θ and hence the probability up to error ε in

$$\mathcal{O}\left(\frac{1}{\varepsilon}\right) \text{ uses of } U$$

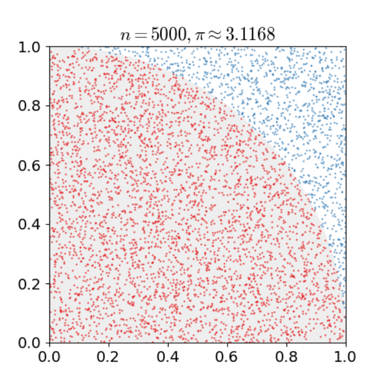
Example: Speeding up Monte Carlo methods

In classical computing a Monte Carlo method samples a lot of times to estimate a probability.



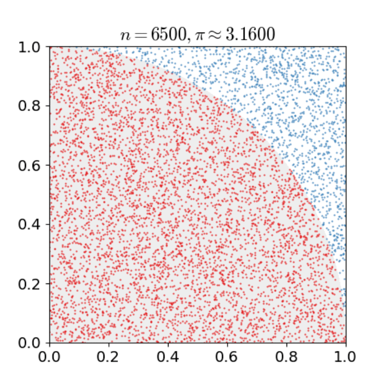
Example: Speeding up Monte Carlo methods

In classical computing a Monte Carlo method samples a lot of times to estimate a probability.



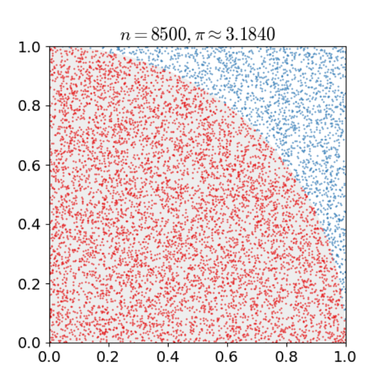
Example: Speeding up Monte Carlo methods

In classical computing a Monte Carlo method samples a lot of times to estimate a probability.



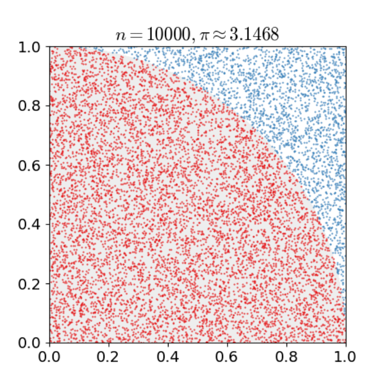
Example: Speeding up Monte Carlo methods

In classical computing a Monte Carlo method samples a lot of times to estimate a probability.



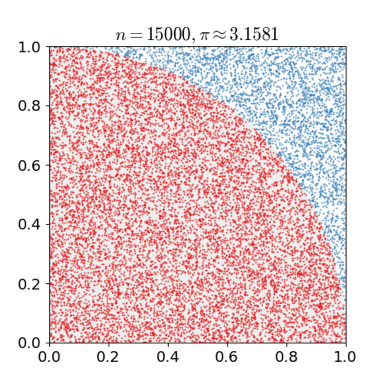
Example: Speeding up Monte Carlo methods

In classical computing a Monte Carlo method samples a lot of times to estimate a probability.



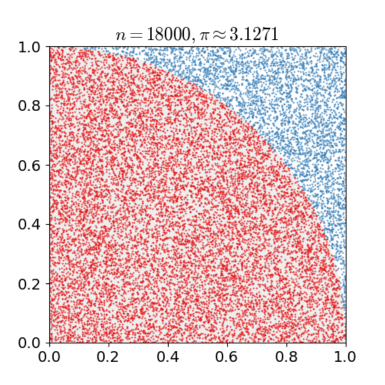
Example: Speeding up Monte Carlo methods

In classical computing a Monte Carlo method samples a lot of times to estimate a probability.



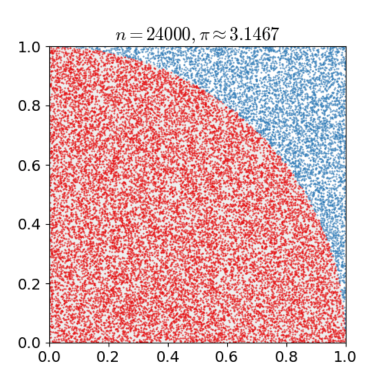
Example: Speeding up Monte Carlo methods

In classical computing a Monte Carlo method samples a lot of times to estimate a probability.



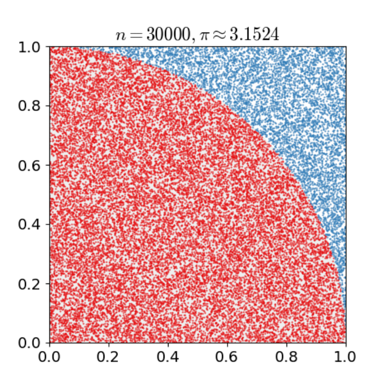
Example: Speeding up Monte Carlo methods

In classical computing a Monte Carlo method samples a lot of times to estimate a probability.



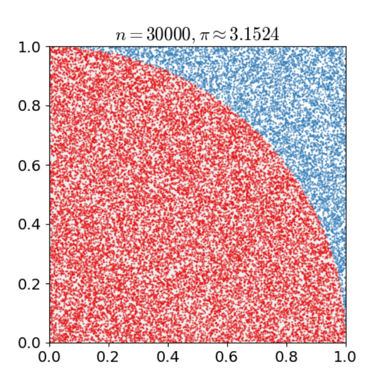
Example: Speeding up Monte Carlo methods

In classical computing a Monte Carlo method samples a lot of times to estimate a probability.



Example: Speeding up Monte Carlo methods

In classical computing a Monte Carlo method samples a lot of times to estimate a probability.



A classical algorithm requires $\sim \frac{1}{\varepsilon^2}$ samples to estimate, quantum $\frac{1}{\varepsilon}$.

Overview of some other algorithms

Some other algorithms

There are many more quantum algorithms, often using the discoed techniques. We will quickly discuss some other important algorithms:

- Shor's algorithm & factoring integers
- HHL for "solving" linear systems
- Faster gradient computation

Factoring

- Problem: given a (large) number N, what are the prime factors?

Factoring

- Problem: given a (large) number N, what are the prime factors?

$$15 =$$

Factoring

- Problem: given a (large) number N, what are the prime factors?

$$15 = 3 \times 5$$

Factoring

- Problem: given a (large) number N, what are the prime factors?

$$15 = 3 \times 5$$

$$391 =$$

Factoring

- Problem: given a (large) number N, what are the prime factors?

$$15 = 3 \times 5$$

$$391 = 17 \times 23$$

Factoring

- Problem: given a (large) number N, what are the prime factors?

$$15 = 3 \times 5$$

$$391 = 17 \times 23$$

- Best known classical algorithms run in <u>exponential time</u> in the number of bits.

Factoring

- Problem: given a (large) number N, what are the prime factors?

$$15 = 3 \times 5$$

$$391 = 17 \times 23$$

- Best known classical algorithms run in exponential time in the number of bits.
- Most modern cryptography assumes this is hard to solve.

Factoring

- Problem: given a (large) number N, what are the prime factors?

$$15 = 3 \times 5$$

$$391 = 17 \times 23$$

- Best known classical algorithms run in <u>exponential time</u> in the number of bits.
- Most modern cryptography assumes this is hard to solve.
- Quantum computers can break this in <u>polynomial time</u>.

Shor's factoring algorithm ('94)

A rough sketch:

Shor's factoring algorithm ('94)

A rough sketch:

- A function f is periodic if $f(a) = f(b) \Leftrightarrow b = a + k$ for some period k.

Shor's factoring algorithm ('94)

A rough sketch:

- A function f is periodic if $f(a) = f(b) \Leftrightarrow b = a + k$ for some period k.
- Classical literature: Factoring is equivalent to period finding for

$$f(x) = a^x \mod N$$

Shor's factoring algorithm ('94)

A rough sketch:

- A function f is periodic if $f(a) = f(b) \Leftrightarrow b = a + k$ for some period k.
- Classical literature: Factoring is equivalent to period finding for

$$f(x) = a^x \mod N$$

- Finding this period is believed to be hard classically.

Shor's factoring algorithm ('94)

A rough sketch:

- A function f is periodic if $f(a) = f(b) \Leftrightarrow b = a + k$ for some period k.
- Classical literature: Factoring is equivalent to period finding for

$$f(x) = a^x \mod N$$

- Finding this period is believed to be hard classically.
- Let U be a unitary such that $U|f(a)\rangle = |f(a)\rangle$.

Shor's factoring algorithm ('94)

A rough sketch:

- A function f is periodic if $f(a) = f(b) \Leftrightarrow b = a + k$ for some period k.
- Classical literature: Factoring is equivalent to period finding for

$$f(x) = a^x \mod N$$

- Finding this period is believed to be hard classically.
- Let U be a unitary such that $U|f(a)\rangle = |f(a)\rangle$.
- Now $U^k|f(a)\rangle = |f(a+k)\rangle = |f(a)\rangle$ so $|f(a)\rangle$ has eigenvalue 1 under U^k.

Shor's factoring algorithm ('94)

A rough sketch:

- A function f is periodic if $f(a) = f(b) \Leftrightarrow b = a + k$ for some period k.
- Classical literature: Factoring is equivalent to period finding for

$$f(x) = a^x \mod N$$

- Finding this period is believed to be hard classically.
- Let U be a unitary such that $U\,|f(a)\rangle = |f(a)\rangle$.
- Now $U^k\,|f(a)\rangle = |f(a+k)\rangle = |f(a)\rangle$ so $|f(a)\rangle$ has eigenvalue 1 under U^k.
- So $|f(a)\rangle$ is a combination of eigenvectors with eigenvalues $e^{i2\pi\frac{t}{k}}$.

Shor's factoring algorithm ('94)

A rough sketch:

- A function f is periodic if $f(a) = f(b) \Leftrightarrow b = a + k$ for some period k.
- Classical literature: Factoring is equivalent to period finding for

$$f(x) = a^x \mod N$$

- Finding this period is believed to be hard classically.
- Let U be a unitary such that $U|f(a)\rangle = |f(a)\rangle$.
- Now $U^k|f(a)\rangle = |f(a+k)\rangle = |f(a)\rangle$ so $|f(a)\rangle$ has eigenvalue 1 under U^k.
- So $|f(a)\rangle$ is a combination of eigenvectors with eigenvalues $e^{i2\pi\frac{t}{k}}$.
- Phase estimation on U gives a value $\frac{t}{k}$, with a few we can learn k.

Shor's factoring algorithm ('94)

A rough sketch:

- A function f is periodic if $f(a) = f(b) \Leftrightarrow b = a + k$ for some period k.
- Classical literature: Factoring is equivalent to period finding for

$$f(x) = a^x \mod N$$

- Finding this period is believed to be hard classically.
- Let U be a unitary such that $U|f(a)\rangle = |f(a)\rangle$.
- Now $U^k|f(a)\rangle = |f(a+k)\rangle = |f(a)\rangle$ so $|f(a)\rangle$ has eigenvalue 1 under U^k.
- So $|f(a)\rangle$ is a combination of eigenvectors with eigenvalues $e^{i2\pi \frac{t}{k}}$.
- Phase estimation on U gives a value $\frac{t}{k}$, with a few we can learn k.
- Small detail: we can implement U^k faster then k repetition, why?

HHL algorithm for systems of linear equations

An important problem in computer science is linear system solving.

Linear system solving

Given s-sparse matrix $A \in \mathbb{R}^{N \times N}$ and a vector $b \in \mathbb{R}^N$, find a x such that

$$Ax = b$$

HHL algorithm for systems of linear equations

An important problem in computer science is linear system solving.

Linear system solving

Given s-sparse matrix $A \in \mathbb{R}^{N \times N}$ and a vector $b \in \mathbb{R}^N$, find a x such that

$$Ax = b$$

Takes $\widetilde{\mathcal{O}}\left(N^{2.37}\right)$ or $\widetilde{\mathcal{O}}\left(Ns\kappa\right)$ time to solve. $\left(\kappa = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}\right)$

HHL algorithm for systems of linear equations

An important problem in computer science is linear system solving.

Linear system solving

Given s-sparse matrix $A \in \mathbb{R}^{N \times N}$ and a vector $b \in \mathbb{R}^N$, find a x such that

$$Ax = b$$

Takes $\widetilde{\mathcal{O}}\left(N^{2.37}\right)$ or $\widetilde{\mathcal{O}}\left(Ns\kappa\right)$ time to solve. $\left(\kappa = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}\right)$

Quantum linear system solving (HHL '09)

Given s-sparse matrix $A \in \mathbb{R}^{N \times N}$ and way to prepare $|b\rangle \in \mathbb{R}^N$, prepare a state $|x\rangle$ such that

$$A|x\rangle \propto |b\rangle$$

HHL algorithm for systems of linear equations

An important problem in computer science is linear system solving.

Linear system solving

Given s-sparse matrix $A \in \mathbb{R}^{N \times N}$ and a vector $b \in \mathbb{R}^N$, find a x such that

$$Ax = b$$

Takes $\widetilde{\mathcal{O}}\left(N^{2.37}\right)$ or $\widetilde{\mathcal{O}}\left(Ns\kappa\right)$ time to solve. $\left(\kappa = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}\right)$

Quantum linear system solving (HHL '09)

Given s-sparse matrix $A \in \mathbb{R}^{N \times N}$ and way to prepare $|b\rangle \in \mathbb{R}^N$, prepare a state $|x\rangle$ such that

$$A|x\rangle \propto |b\rangle$$

Takes only $\widetilde{\mathcal{O}}\left(s\kappa\right)$ time

HHL algorithm for systems of linear equations

An important problem in computer science is linear system solving.

Linear system solving

Given s-sparse matrix $A \in \mathbb{R}^{N\times N}$ and a vector $b \in \mathbb{R}^N$, find a x such that

$$Ax = b$$

Takes $\widetilde{\mathcal{O}}\left(N^{2.37}\right)$ or $\widetilde{\mathcal{O}}\left(Ns\kappa\right)$ time to solve. $\left(\kappa = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}\right)$

Quantum linear system solving (HHL '09)

Given s-sparse matrix $A \in \mathbb{R}^{N\times N}$ and way to prepare $|b\rangle \in \mathbb{R}^N$, prepare a state $|x\rangle$ such that

$$A|x\rangle \propto |b\rangle$$

Takes only $\widetilde{\mathcal{O}}\left(s\kappa\right)$ time, but is not the same problem!

Gradient computation (Jordan '05)

Gradient computation

Given black-box access to a function $f\colon \mathbb{R}^n \to \mathbb{R}$, estimate the gradient at 0.

Gradient computation (Jordan '05)

Gradient computation

Given black-box access to a function $f\colon \mathbb{R}^n \to \mathbb{R}$, estimate the gradient at 0.

- Classical algorithm: "walk" a bit in each direction and look at the difference: $\mathcal{O}(n)$ queries.

Gradient computation (Jordan '05)

Gradient computation

Given black-box access to a function $f\colon \mathbb{R}^n \to \mathbb{R}$, estimate the gradient at 0.

- Classical algorithm: "walk" a bit in each direction and look at the difference: $\mathcal{O}(n)$ queries.
- Quantum can do better!

Gradient computation (Jordan '05)

Gradient computation

Given black-box access to a function $f : \mathbb{R}^n \to \mathbb{R}$, estimate the gradient at 0.

- Classical algorithm: "walk" a bit in each direction and look at the difference: $\mathcal{O}(n)$ queries.
- Quantum can do better!
- Assume f is (close to) linear: $f(x) \approx \sum_{i=1}^{n} a_i x_i$, want the a_i's.

Gradient computation (Jordan '05)

Gradient computation

Given black-box access to a function $f\colon \mathbb{R}^n \to \mathbb{R}$, estimate the gradient at 0.

- Classical algorithm: "walk" a bit in each direction and look at the difference: $\mathcal{O}(n)$ queries.
- Quantum can do better!
- Assume f is (close to) linear: $f(x) \approx \sum_{i=1}^{n} a_i x_i$, want the a_i's.
- Idea: query $f(x)$ in the phase

$$|x_1\rangle|\ldots\rangle|x_n\rangle \mapsto e^{i2\pi f(x)}\,|x_1\rangle|\ldots\rangle|x_n\rangle$$

Gradient computation (Jordan '05)

Gradient computation

Given black-box access to a function $f\colon \mathbb{R}^n \to \mathbb{R}$, estimate the gradient at 0.

- Classical algorithm: "walk" a bit in each direction and look at the difference: $\mathcal{O}(n)$ queries.
- Quantum can do better!
- Assume f is (close to) linear: $f(x) \approx \sum_{i=1}^{n} a_i x_i$, want the a_i's.
- Idea: query $f(x)$ in the phase

$$|x_1\rangle|\ldots\rangle|x_n\rangle \mapsto e^{i2\pi f(x)}\,|x_1\rangle|\ldots\rangle|x_n\rangle = e^{\sum_{i=1}^{n} a_i x_i}\,|x_1\rangle|\ldots\rangle|x_n\rangle$$

Gradient computation (Jordan '05)

Gradient computation

Given black-box access to a function $f\colon \mathbb{R}^n \to \mathbb{R}$, estimate the gradient at 0.

- Classical algorithm: "walk" a bit in each direction and look at the difference: $\mathcal{O}(n)$ queries.
- Quantum can do better!
- Assume f is (close to) linear: $f(x) \approx \sum_{i=1}^{n} a_i x_i$, want the a_i's.
- Idea: query $f(x)$ in the phase

$$|x_1\rangle|\ldots\rangle|x_n\rangle \mapsto e^{i2\pi f(x)} |x_1\rangle|\ldots\rangle|x_n\rangle = e^{\sum_{i=1}^{n} a_i x_i} |x_1\rangle|\ldots\rangle|x_n\rangle$$

$$= e^{a_1 x_1} |x_1\rangle \ldots e^{a_n x_n} |x_n\rangle$$

Gradient computation (Jordan '05)

Gradient computation

Given black-box access to a function $f\colon \mathbb{R}^n \to \mathbb{R}$, estimate the gradient at 0.

- Classical algorithm: "walk" a bit in each direction and look at the difference: $\mathcal{O}(n)$ queries.
- Quantum can do better!
- Assume f is (close to) linear: $f(x) \approx \sum_{i=1}^{n} a_i x_i$, want the a_i's.
- Idea: query $f(x)$ in the phase

$$|x_1\rangle|\ldots\rangle|x_n\rangle \mapsto e^{i2\pi f(x)}\,|x_1\rangle|\ldots\rangle|x_n\rangle = e^{\sum_{i=1}^{n} a_i x_i}\,|x_1\rangle|\ldots\rangle|x_n\rangle$$

$$= e^{a_1 x_1}\,|x_1\rangle \ldots e^{a_n x_n}\,|x_n\rangle$$

- Use phase estimation on all n coordinates at the same time!

A quiz

By show of hand, true or false?

A quiz

By show of hand, true or false?

1. Quantum computers are exponentially faster at search problems than classical computers because they can try everything in superposition.

A quiz

By show of hand, true or false?

1. Quantum computers are exponentially faster at search problems than classical computers because they can try everything in superposition.
2. Quantum computers are exponentially faster at some problems.

A quiz

By show of hand, true or false?

1. Quantum computers are exponentially faster at search problems than classical computers because they can try everything in superposition.
2. Quantum computers are exponentially faster at some problems.
3. Quantum computers are quadratically faster at many problems that include sampling.

A quiz

By show of hand, true or false?

1. Quantum computers are exponentially faster at search problems than classical computers because they can try everything in superposition.
2. Quantum computers are exponentially faster at some problems.
3. Quantum computers are quadratically faster at many problems that include sampling.
4. Quantum computers can break most modern cryptography.

A quiz

By show of hand, true or false?

1. Quantum computers are exponentially faster at search problems than classical computers because they can try everything in superposition.
2. Quantum computers are exponentially faster at some problems.
3. Quantum computers are quadratically faster at many problems that include sampling.
4. Quantum computers can break most modern cryptography.
5. Quantum computers can solve linear systems of equations faster.

A quiz

By show of hand, true or false?

1. Quantum computers are exponentially faster at search problems than classical computers because they can try everything in superposition.
2. Quantum computers are exponentially faster at some problems.
3. Quantum computers are quadratically faster at many problems that include sampling.
4. Quantum computers can break most modern cryptography.
5. Quantum computers can solve linear systems of equations faster.
6. Quantum computers are wierd but cool.

That was it!