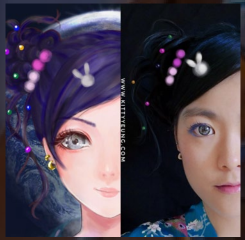
Grover’s Algorithm

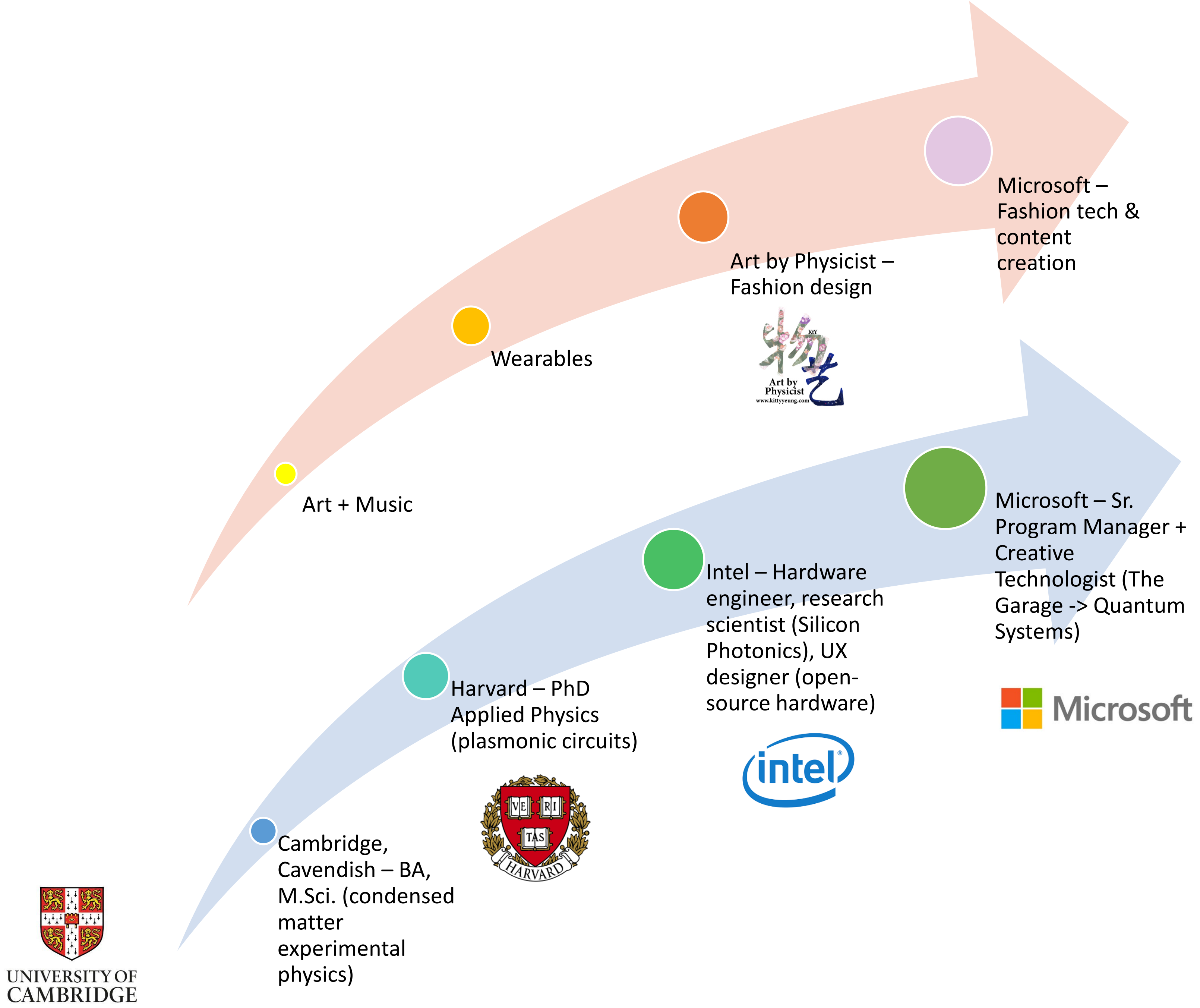


Kitty Yeung, Ph.D. in Applied Physics

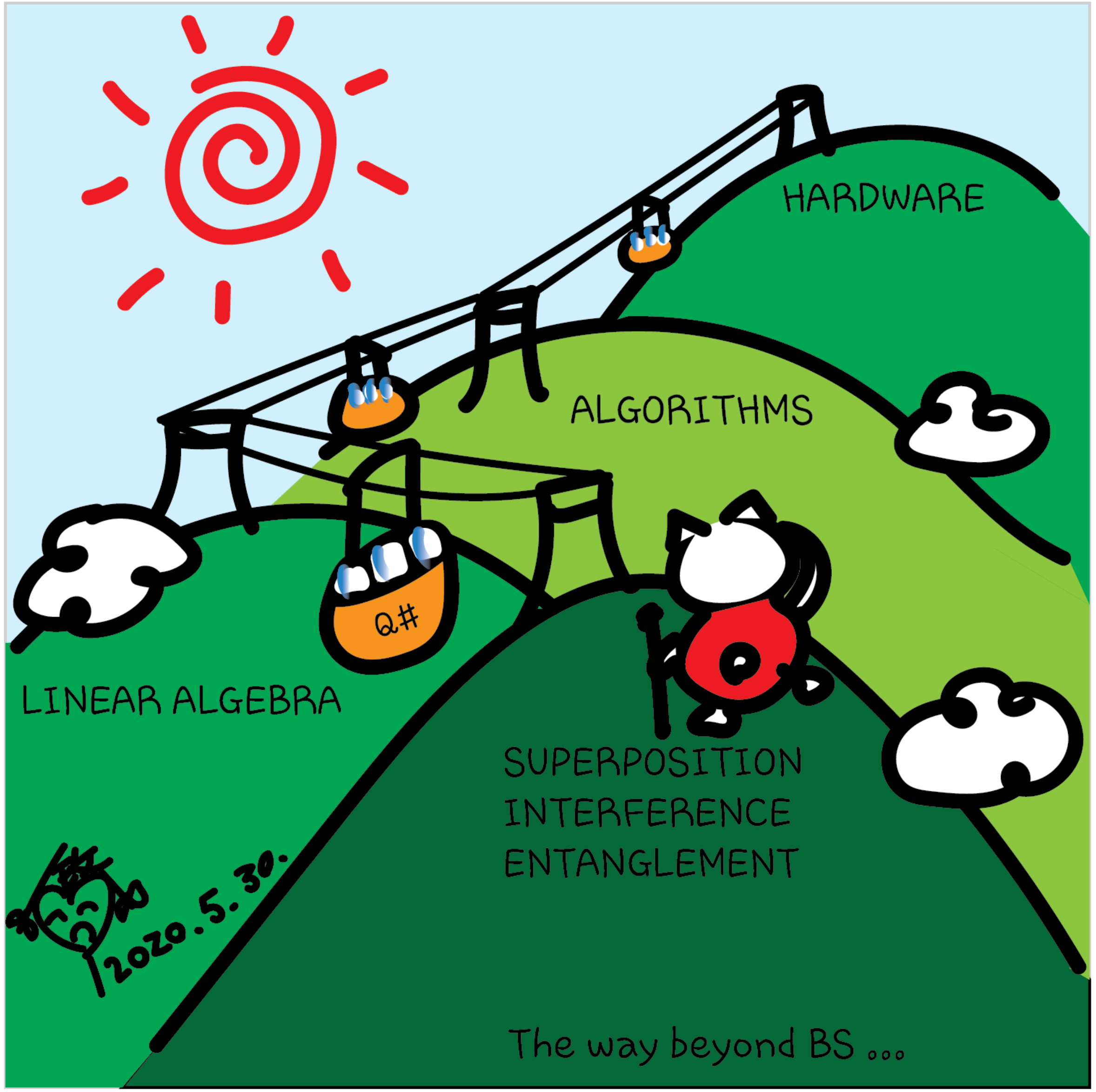
Creative Technologist + Sr. PM
Microsoft

www.artbyphysicistkittyyeung.com
@KittyArtPhysics
@artbyphysicistkittyyeung

Oct 3, 2020
Zen4Quantum Meetup

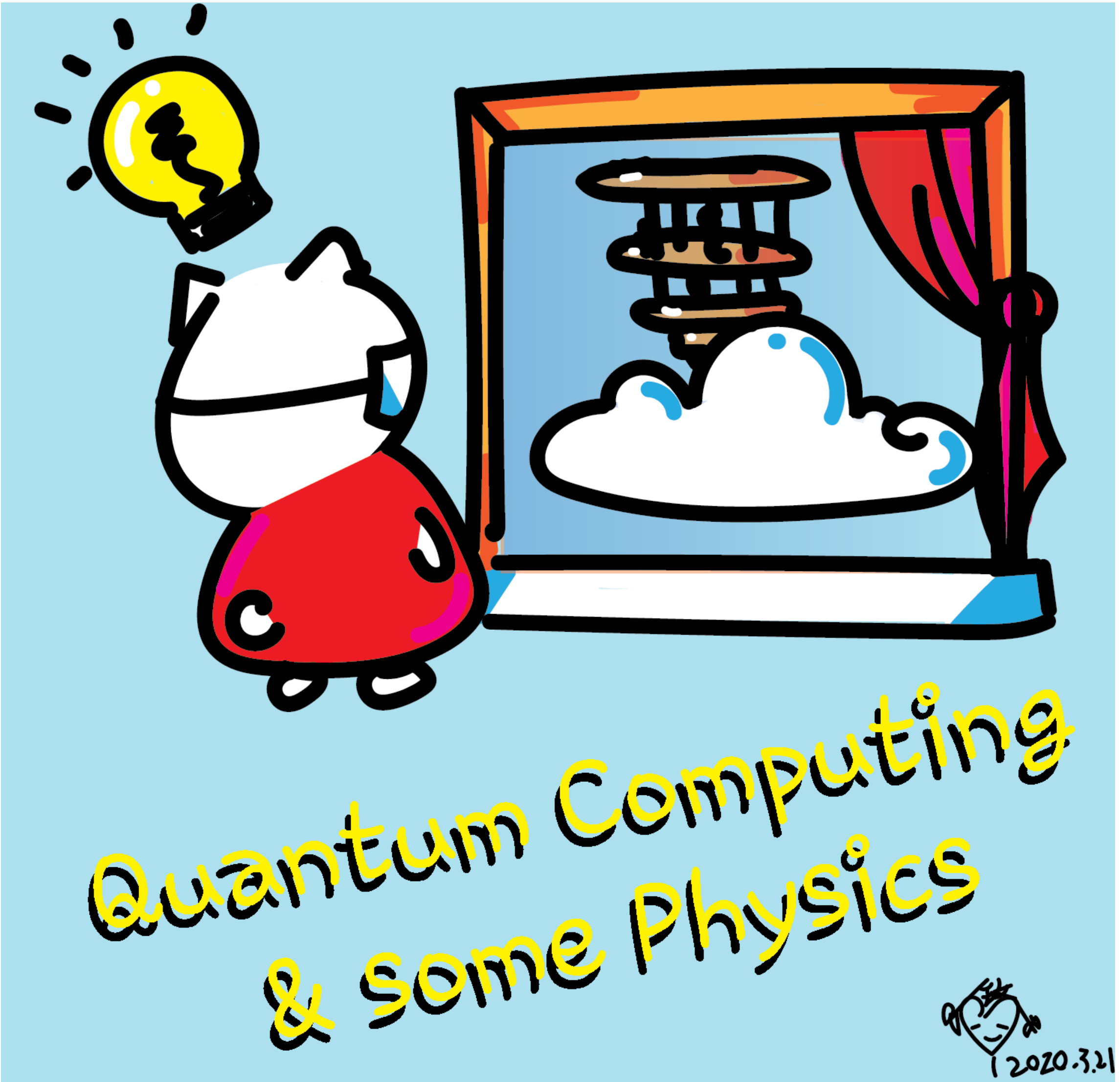


Art + Music
Wearables
Art by Physicist – Fashion design
Art by Physicist
www.kittyyeung.com
Microsoft – Fashion tech & content creation
UNIVERSITY OF CAMBRIDGE
Cambridge, Cavendish – BA, M.Sci. (condensed matter experimental physics)
Harvard – PhD Applied Physics (plasmonic circuits)
VE RI TAS
HARVARD
Intel – Hardware engineer, research scientist (Silicon Photonics), UX designer (open-source hardware)
intel
Microsoft – Sr. Program Manager + Creative Technologist (The Garage -> Quantum Systems)
Microsoft


HARDWARE
ALGORITHMS
Q#
LINEAR ALGEBRA
SUPERPOSITION
INTERFERENCE
ENTANGLEMENT
2020.5.30.
The way beyond BS ...

Class structure

- [Comics on Hackaday – Introduction to Quantum Computing]() every Sun
- 30 mins – 1 hour every Sun, one concept (theory, hardware, programming), Q&A
- Contribute to Q# documentation http://docs.microsoft.com/quantum
- Coding through Quantum Katas https://github.com/Microsoft/QuantumKatas/
- Discuss in Hackaday project comments throughout the week
- Take notes

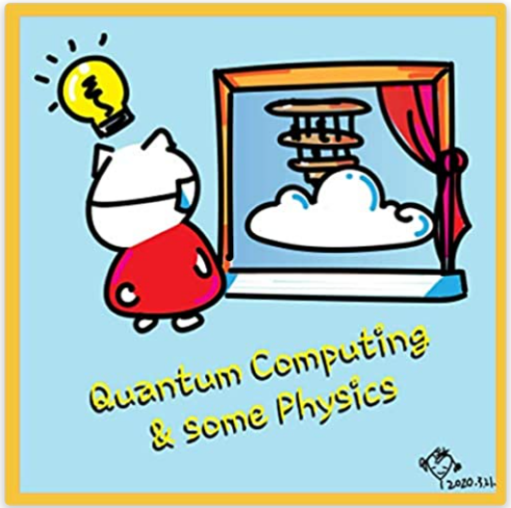


ASIN : B08HGLPZXP in 13 markets

Books | Advanced Search | New Releases | Best Sellers & More | Children's Books | Textbooks | Textbook Rentals | Magazines | Best Books of the Month

Books › Comics & Graphic Novels › Graphic Novels

Look inside



Quantum Computing & Some Physics: The Quantum Computing Comics Notebook

Paperback – September 3, 2020

by Dr. Kitty Yeung (Author)

See all formats and editions

Paperback
$19.98

1 New from $19.98

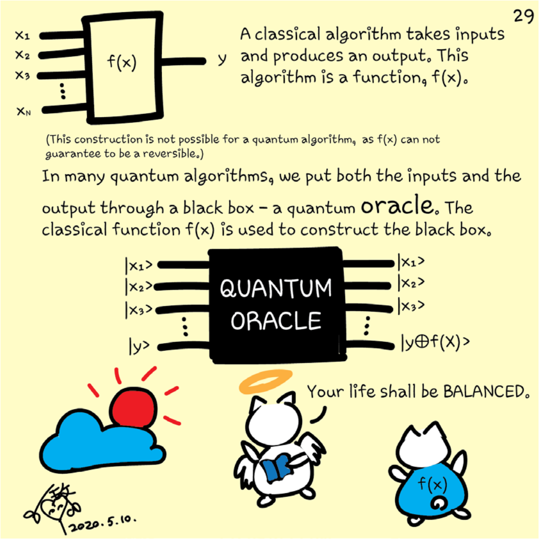
Learn about quantum computing through an intuitive series of comics. It is both a book and a notebook, in which readers can note down their thoughts on the back of the comics. The book provides a high-level guide to the basic concepts of quantum computing, linear algebra, and quantum algorithms. Commonly used quantum hardware architectures are also described in the comics. Learners at any age with any background can get something out of this comics. The

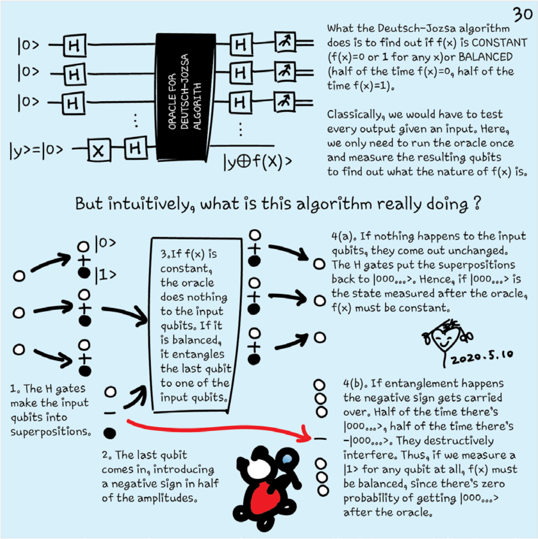
Quantum Algorithms

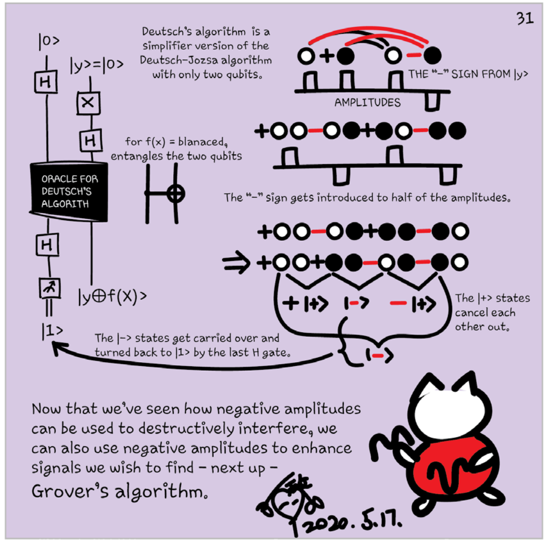
Performing calculations based on the laws of quantum mechanics

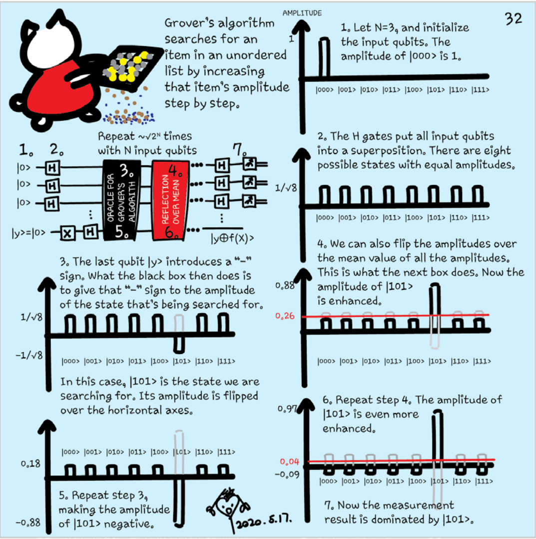
- 1980 & 1982: Manin & Feynman proposed the idea of creating machines based on the laws of quantum mechanics
- 1985: David Deutsch developed Quantum Turing machine, showing that quantum circuits are universal
- 1994: Peter Shor came up with a quantum algorithm to factor very large numbers in polynomial time
- 1997: Grover developed a quantum search algorithm with $O(\sqrt{N})$ complexity

Quantum algorithms leverage superposition, interference and entanglement



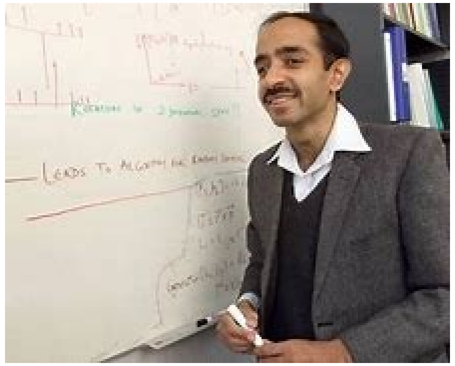




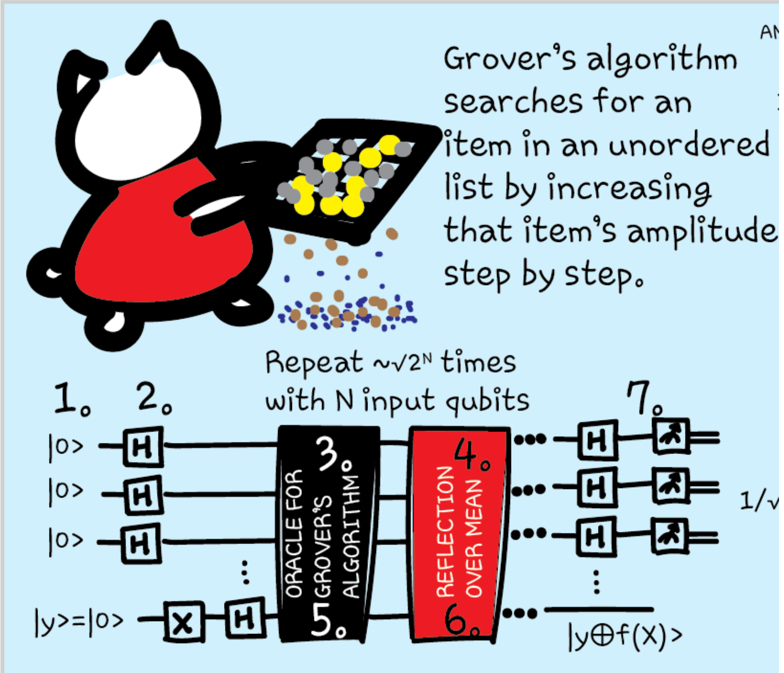


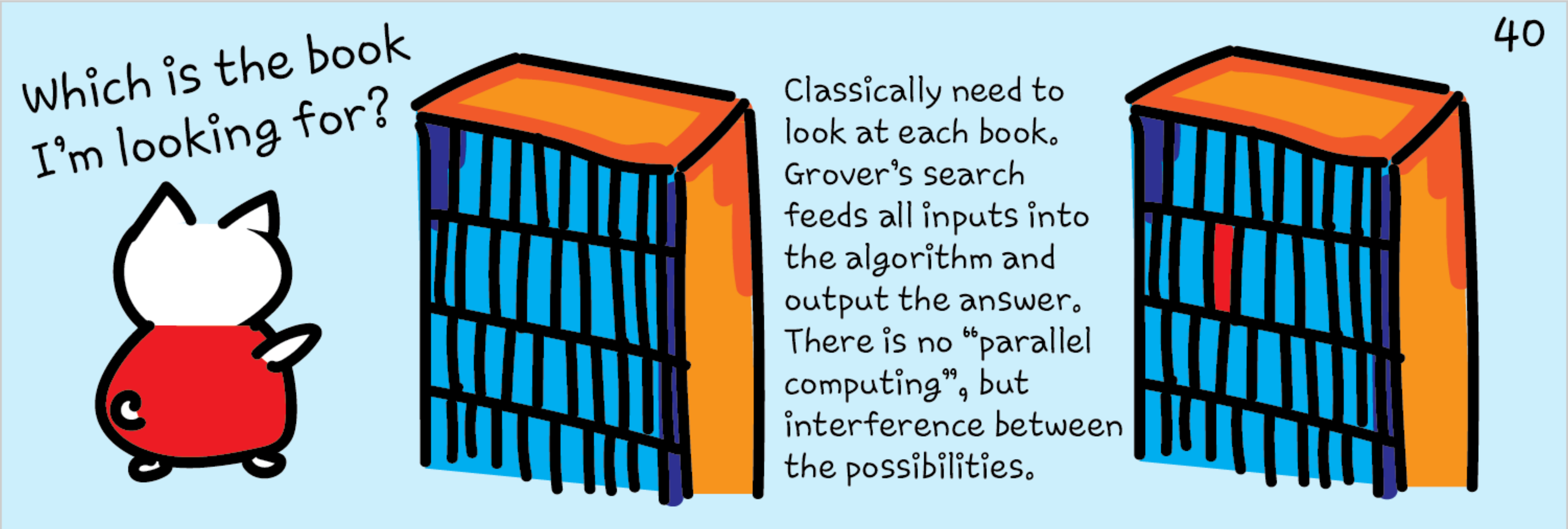
Grover’s algorithm

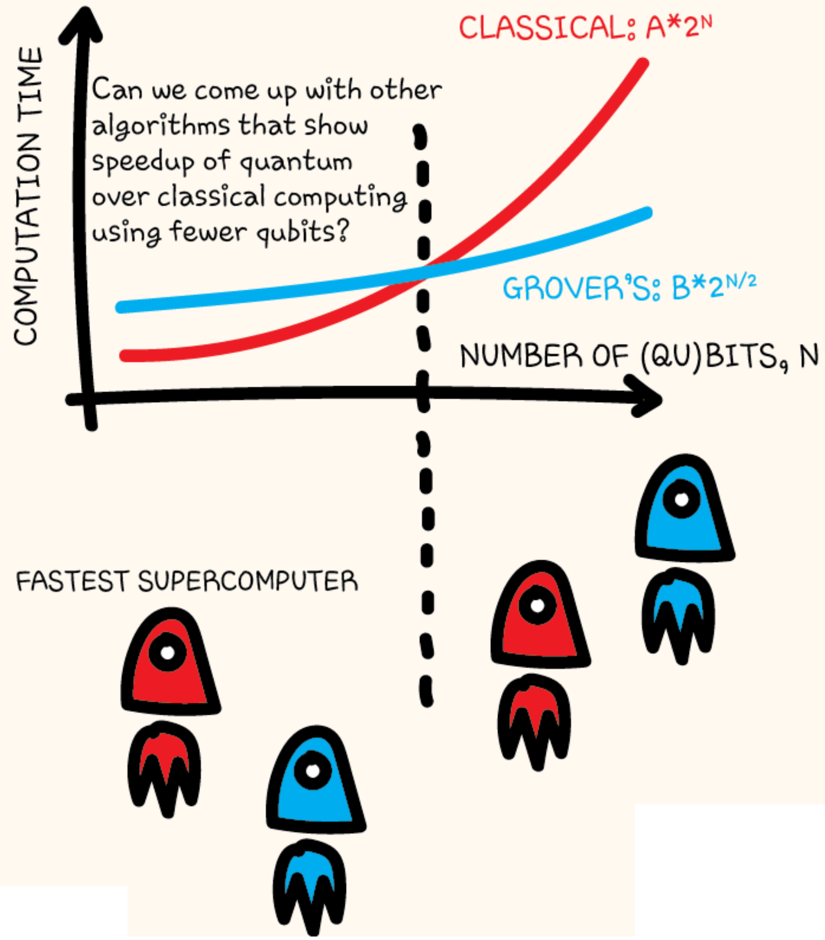
https://en.wikipedia.org/wiki/Grover%27s_algorithm



Lov Kumar Grover (* 1960 in Merath, India) is an Indian-American computer scientist




Which is the book I'm looking for?
Classically need to look at each book. Grover's search feeds all inputs into the algorithm and output the answer. There is no "parallel computing", but interference between the possibilities.



Grover's algorithm provides a speedup over classical algorithms for searching for an item in an unordered list (after a certain number of qubits are reached). A and B are factors that don't depend on N. (They describe how long it takes for the computers to complete the task for a fixed N).

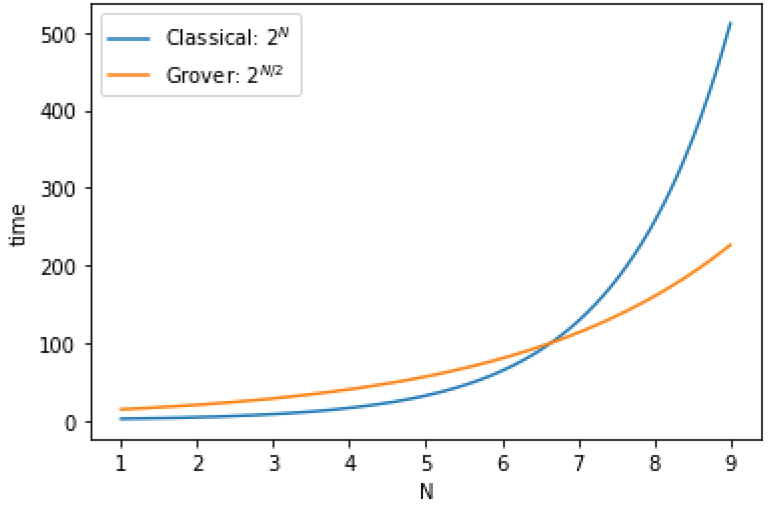
The reflection over the |000...0> state surrounded by H gates is the reflection over mean.

```
[ ] import matplotlib.pyplot as plt
    import numpy as np
```

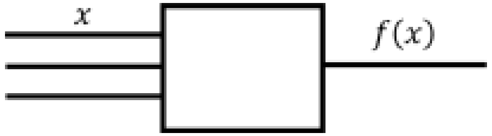
```
[ ] x = np.linspace(1, 9, 1000)
    y = 2**x
    z = 10 * np.sqrt(2**x)
```

```
[ ] plt.plot(x, y, label=r'Classical: $2^N$')
    plt.plot(x, z, label=r'Grover: $2^{N/2}$')
    plt.legend()
    plt.xlabel('N')
    plt.ylabel('time')
```

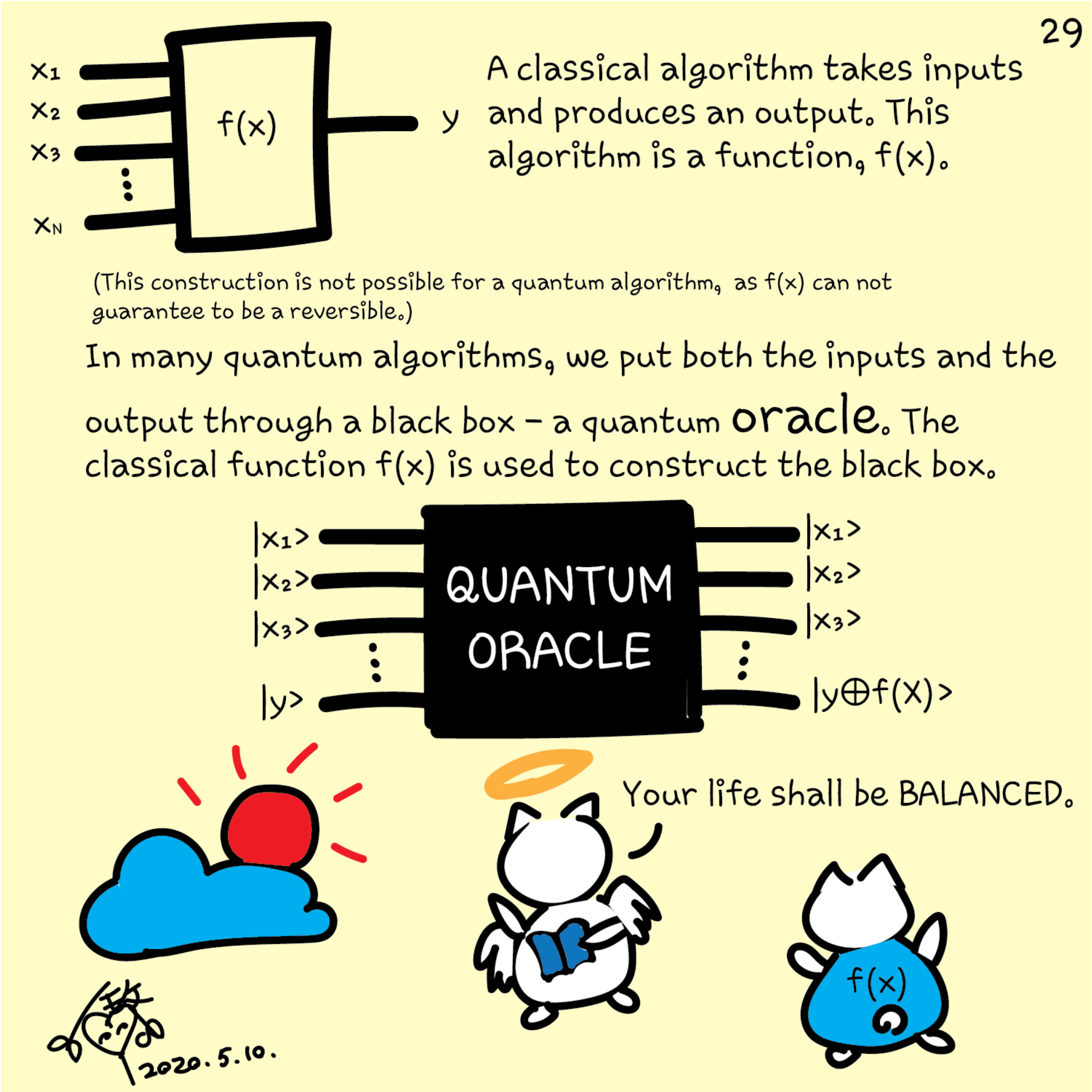
```
Text(0, 0.5, 'time')
```



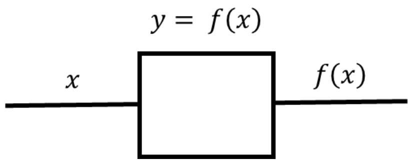
(May 10 Session 7)



x	$y = f(x)$
000	0
001	0
010	0
011	0
100	0
101	0
110	1
111	0



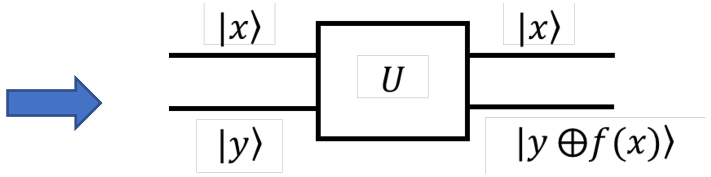
Oracles



x	$y = f(x) = x \% 4$
0	0
1	1
2	2
3	3
4	0
5	1
6	2
7	3

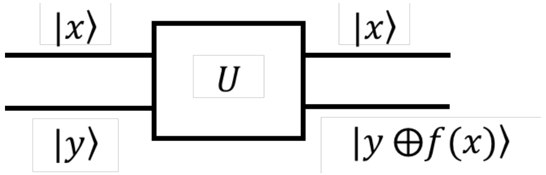
Quantum needs unitary gates = reversible

$$A^{\dagger}(y) = A^{-1}(y) = A^{-1}(Ax) = x$$



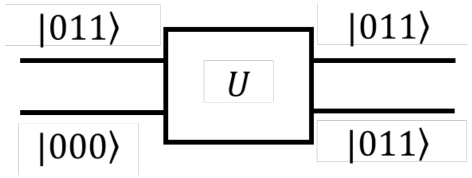
Cannot exist for circuit on the left

Oracles

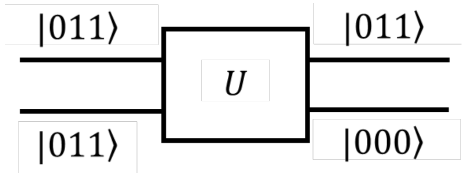


x	$y = f(x) = x \,\%\, 4$
0	0
1	1
2	2
3	3
4	0
5	1
6	2
7	3

for $x = 3 = |011\rangle$, with y initialized to $0 = |000\rangle$

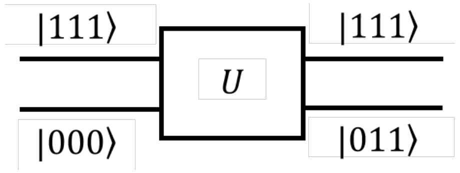


$U(|011\rangle, |000\rangle) = (|011\rangle, |000\rangle \oplus f(|011\rangle)) = (|011\rangle, |000\rangle \oplus |011\rangle) = (|011\rangle, |011\rangle)$

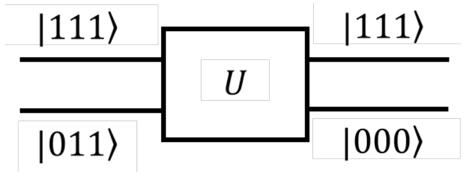


$U(|111\rangle, |000\rangle) = (|111\rangle, |000\rangle \oplus f(|111\rangle)) = (|111\rangle, |000\rangle \oplus |011\rangle) = (|111\rangle, |011\rangle)$

for $x = 7 = |111\rangle$, with y initialized to $0 = |000\rangle$

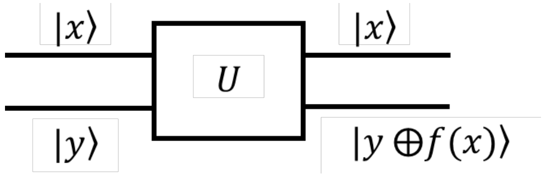


$U(|111\rangle, |000\rangle) = (|111\rangle, |000\rangle \oplus f(|111\rangle)) = (|111\rangle, |000\rangle \oplus |011\rangle) = (|111\rangle, |011\rangle)$



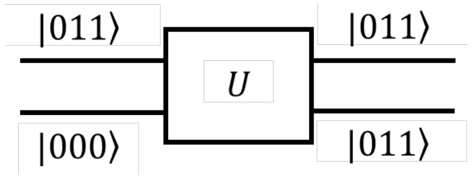
$U(|111\rangle, |011\rangle) = (|111\rangle, |011\rangle \oplus f(|111\rangle)) = (|111\rangle, |011\rangle \oplus |011\rangle) = (|111\rangle, |000\rangle)$

Oracles

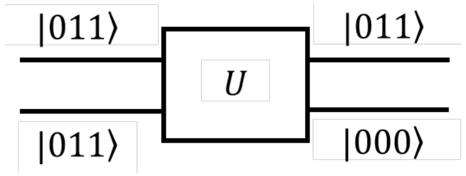


x	$y = f(x) = x \% 4$
0	0
1	1
2	2
3	3
4	0
5	1
6	2
7	3

for $x = 3 = |011\rangle$, with y initialized to $0 = |000\rangle$

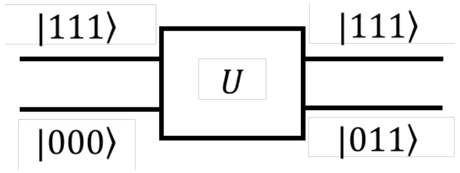


$U(|011\rangle, |000\rangle) = (|011\rangle, |000\rangle \oplus f(|011\rangle)) = (|011\rangle, |000\rangle \oplus |011\rangle) = (|011\rangle, |011\rangle)$

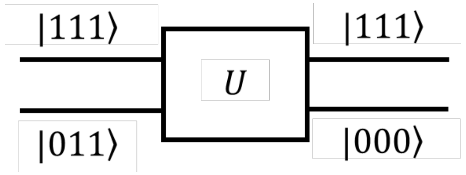


$U(|111\rangle, |000\rangle) = (|111\rangle, |000\rangle \oplus f(|111\rangle)) = (|111\rangle, |000\rangle \oplus |011\rangle) = (|111\rangle, |011\rangle)$

for $x = 7 = |111\rangle$, with y initialized to $0 = |000\rangle$

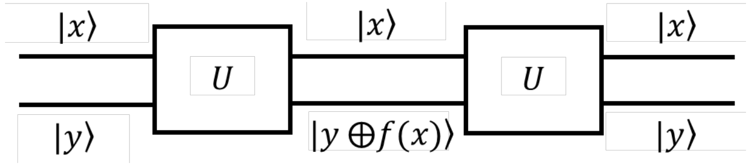


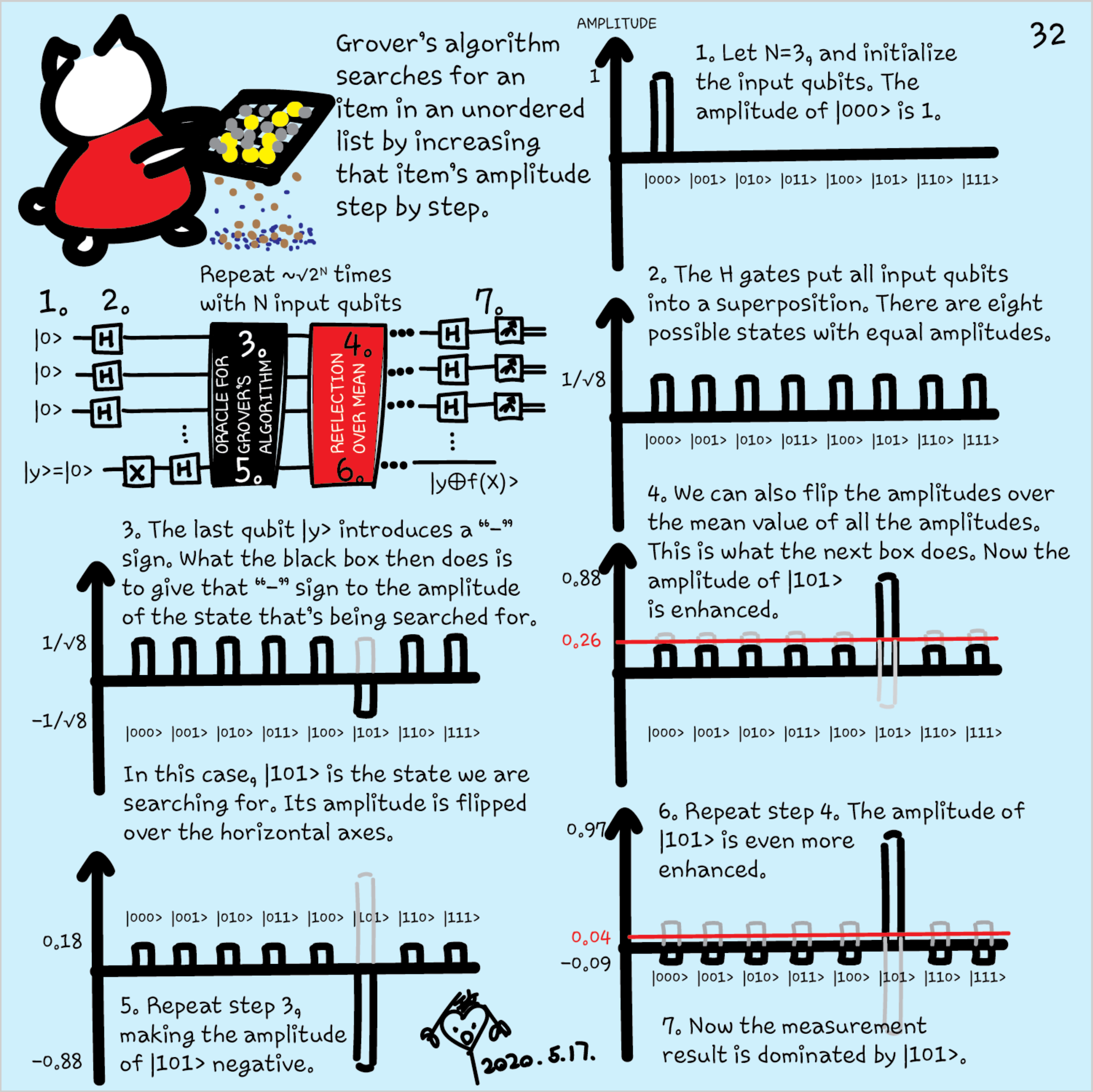
$U(|111\rangle, |000\rangle) = (|111\rangle, |000\rangle \oplus f(|111\rangle)) = (|111\rangle, |000\rangle \oplus |011\rangle) = (|111\rangle, |011\rangle)$



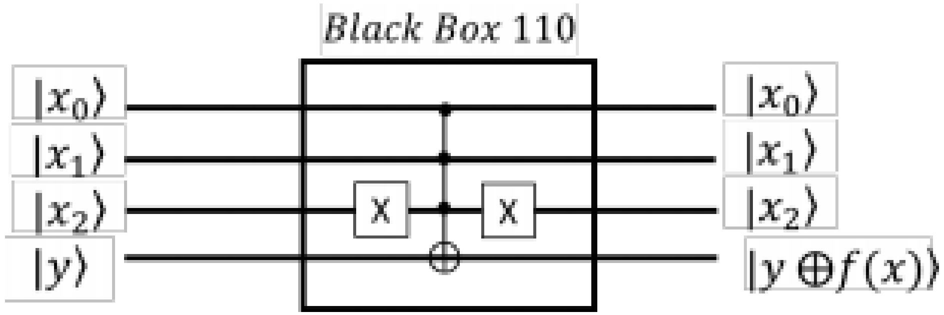
$U(|111\rangle, |011\rangle) = (|111\rangle, |011\rangle \oplus f(|111\rangle)) = (|111\rangle, |011\rangle \oplus |011\rangle) = (|111\rangle, |000\rangle)$

$$U = U^{-1}$$

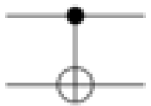



32
Grover's algorithm searches for an item in an unordered list by increasing that item's amplitude step by step.
AMPLITUDE
1
1. Let N=3, and initialize the input qubits. The amplitude of |000> is 1.
|000> |001> |010> |011> |100> |101> |110> |111>
Repeat ~√2^N times with N input qubits
1. 2.
|0>
|0>
|0>
|y>=|0>
H
X
3. ORACLE FOR GROVER'S ALGORITHM 5.
4. REFLECTION OVER MEAN 6.
7.
|y⊕f(X)>
2. The H gates put all input qubits into a superposition. There are eight possible states with equal amplitudes.
1/√8
|000> |001> |010> |011> |100> |101> |110> |111>
3. The last qubit |y> introduces a "–" sign. What the black box then does is to give that "–" sign to the amplitude of the state that's being searched for.
1/√8
-1/√8
|000> |001> |010> |011> |100> |101> |110> |111>
In this case, |101> is the state we are searching for. Its amplitude is flipped over the horizontal axes.
4. We can also flip the amplitudes over the mean value of all the amplitudes. This is what the next box does. Now the amplitude of |101> is enhanced.
0.88
0.26
|000> |001> |010> |011> |100> |101> |110> |111>
0.18
-0.88
|000> |001> |010> |011> |100> |101> |110> |111>
5. Repeat step 3, making the amplitude of |101> negative.
2020.5.17.
6. Repeat step 4. The amplitude of |101> is even more enhanced.
0.97
0.04
-0.09
|000> |001> |010> |011> |100> |101> |110> |111>
7. Now the measurement result is dominated by |101>.

x	$y = f(x)$
000	0
001	0
010	0
011	0
100	0
101	0
110	1
111	0

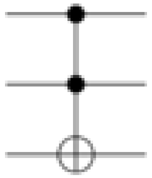


Controlled Not (CNOT, CX)

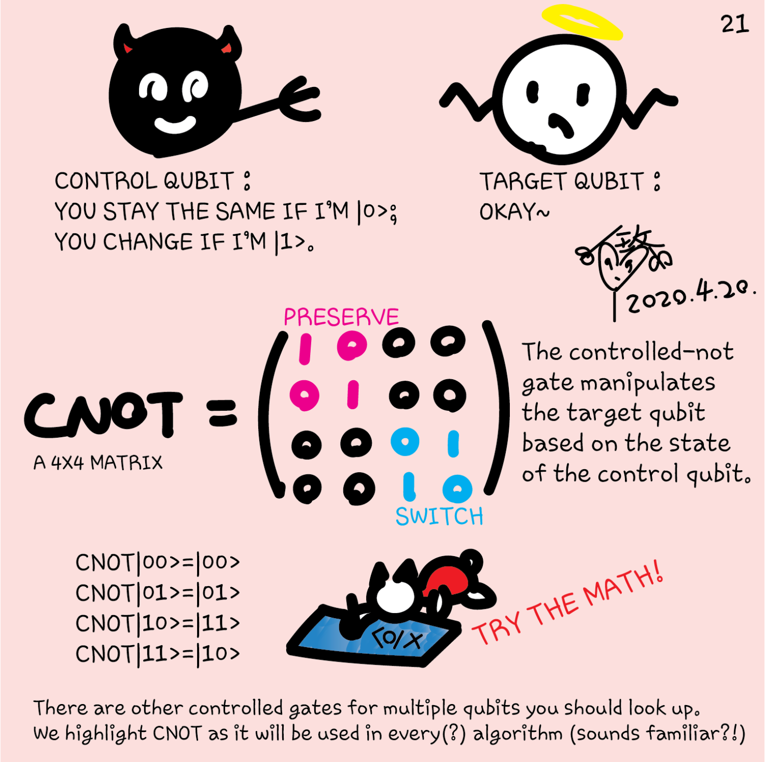


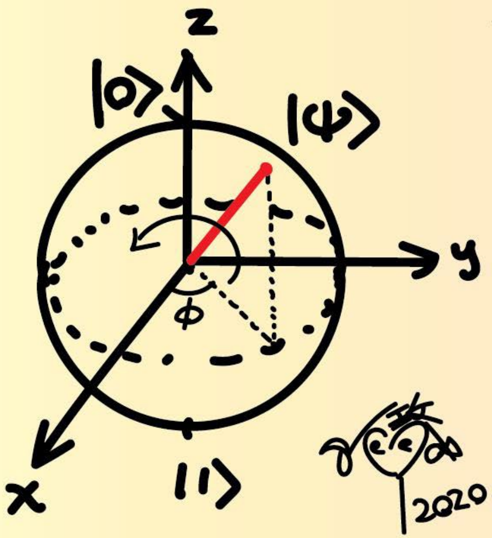
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Toffoli (CCNOT, CCX, TOFF)



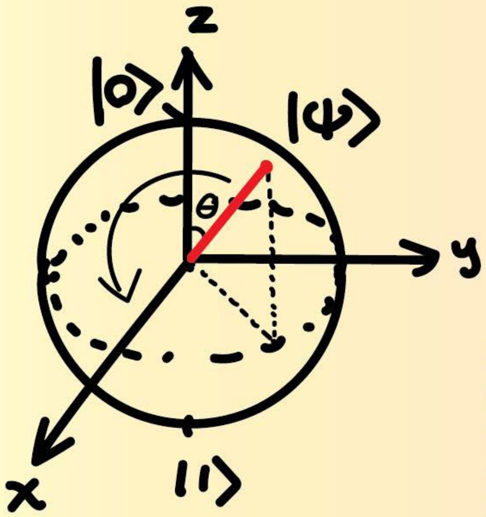
$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$





To change the phase φ, we have a commonly used gate, Z, which rotates about the z-axis by 180°.

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

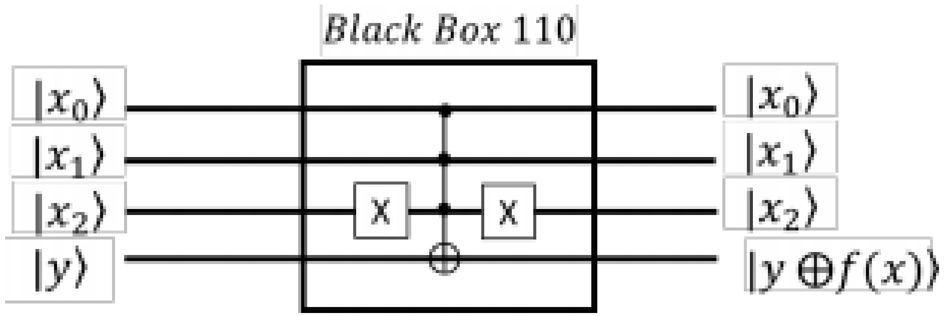


Similarly, the X gate rotates about the x-axis by 180°, rotating the angle θ e.g. $X|0\rangle = |1\rangle$, $X|1\rangle = |0\rangle$.

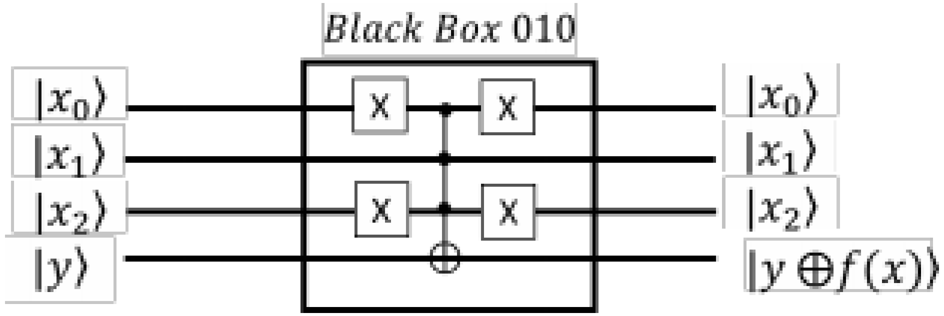
$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

We have seen in page 18 the two matrices for changing φ and θ in arbitraty amounts. They form a universal gate set - they can put a state anywhere on the Bloch Sphere. The gates Z and X are special cases of them.

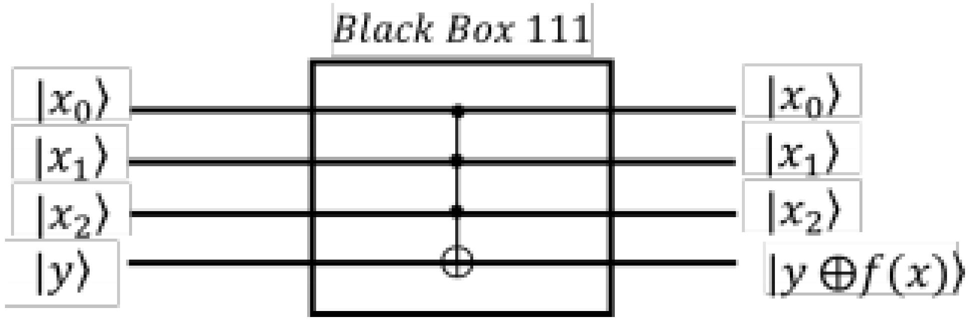
x	$y = f(x)$
000	0
001	0
010	0
011	0
100	0
101	0
110	1
111	0

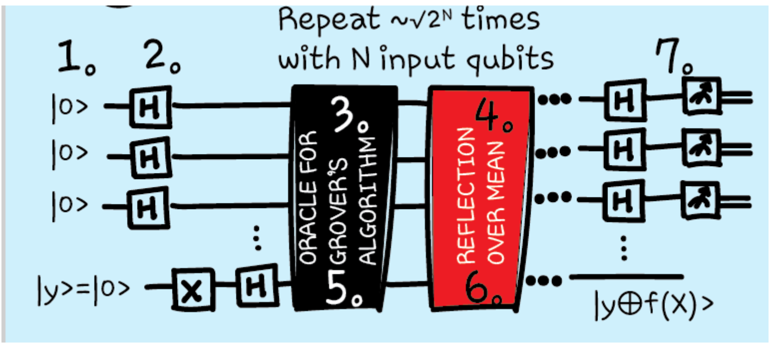


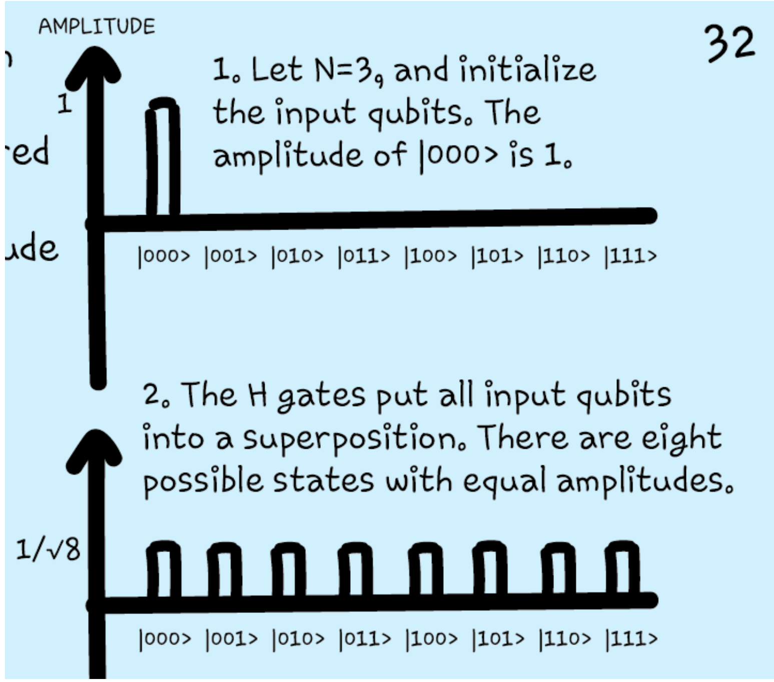
x	$y = f(x)$
000	0
001	0
010	1
011	0
100	0
101	0
110	0
111	0

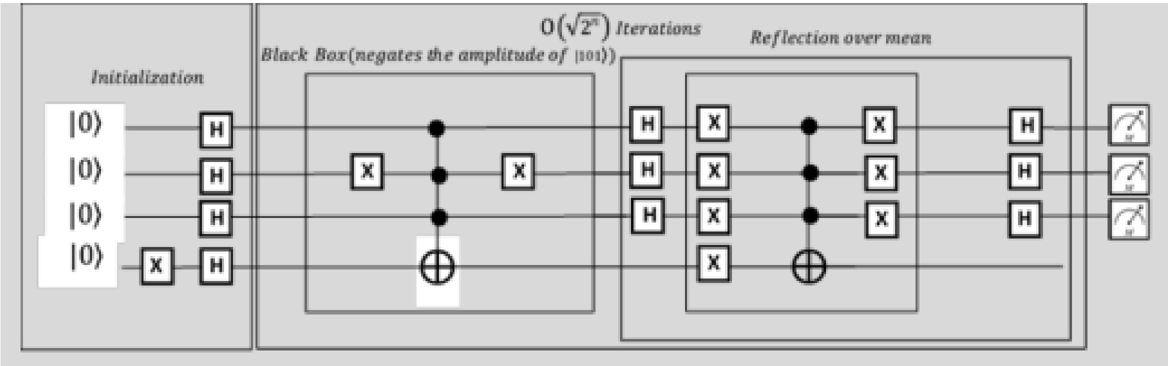


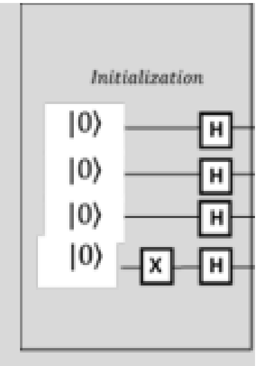
x	$y = f(x)$
000	0
001	0
010	0
011	0
100	0
101	0
110	0
111	1

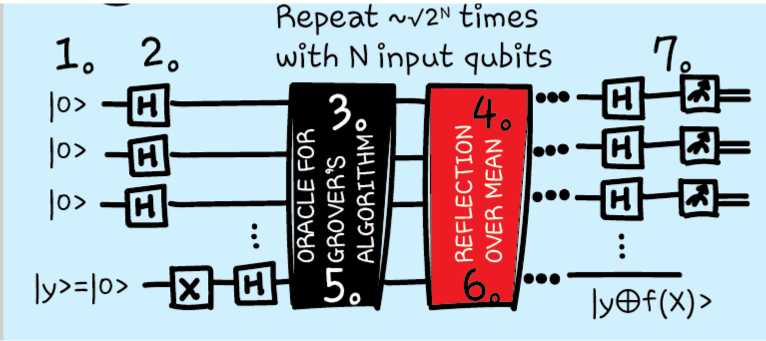


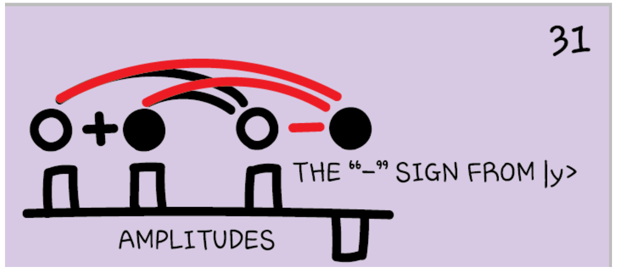

Repeat ~√2ᴺ times
with N input qubits
1.
2.
7.
|0>
|0>
|0>
|y>=|0>
H
X
3.
ORACLE FOR GROVER'S ALGORITHM
5.
4.
REFLECTION OVER MEAN
6.
|y⊕f(X)>


AMPLITUDE
32
1
1. Let N=3, and initialize the input qubits. The amplitude of |000> is 1.
|000> |001> |010> |011> |100> |101> |110> |111>
2. The H gates put all input qubits into a superposition. There are eight possible states with equal amplitudes.
1/√8
|000> |001> |010> |011> |100> |101> |110> |111>

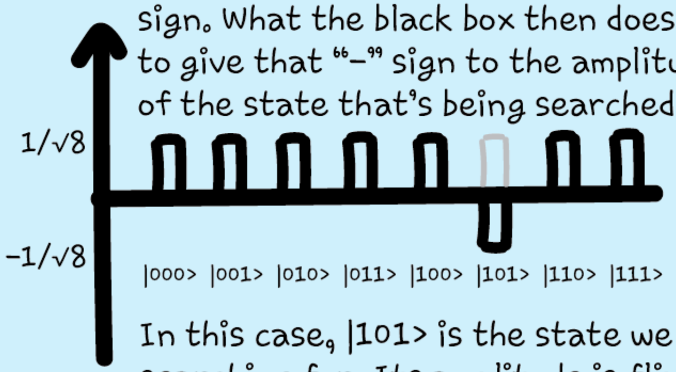

O(√2ⁿ) Iterations
Initialization
Black Box(negates the amplitude of |101⟩)
Reflection over mean
|0⟩
|0⟩
|0⟩
|0⟩
H
X


Initialization
|0⟩
|0⟩
|0⟩
|0⟩
X
H

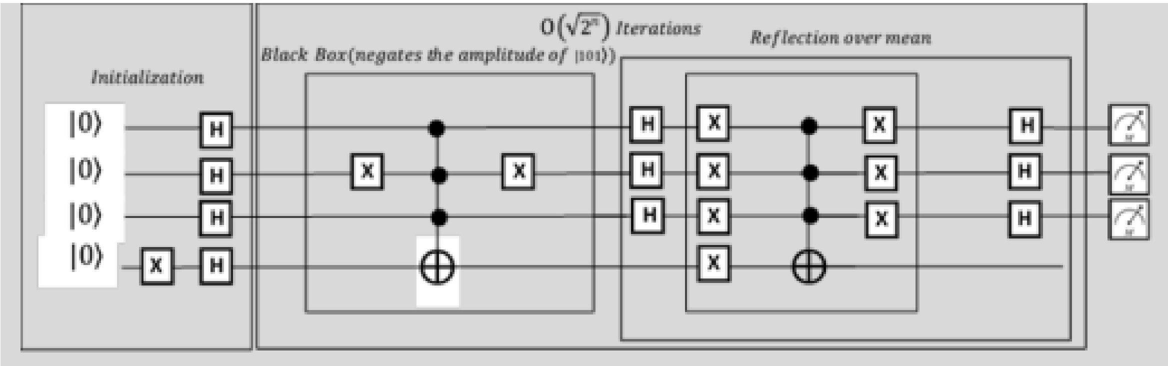


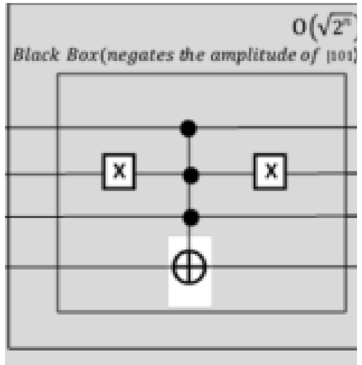


3. The last qubit |y> introduces a "–" sign. What the black box then does is to give that "–" sign to the amplitude of the state that's being searched for.



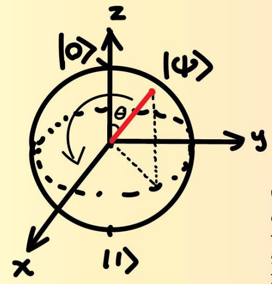
In this case, |101> is the state we are searching for. Its amplitude is flipped over the horizontal axes.





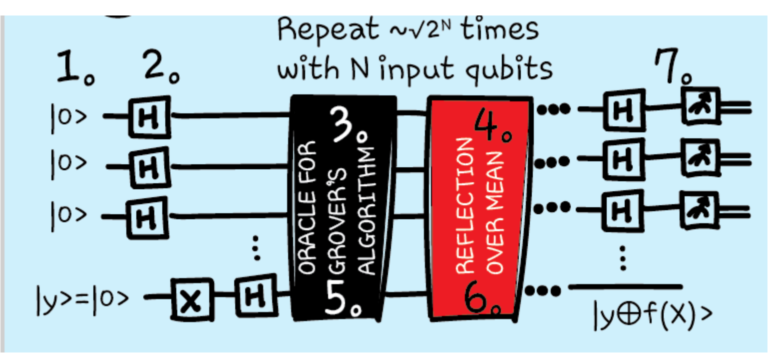
x	$y = f(x)$
000	0
001	0
010	0
011	0
100	0
101	1
110	0
111	0

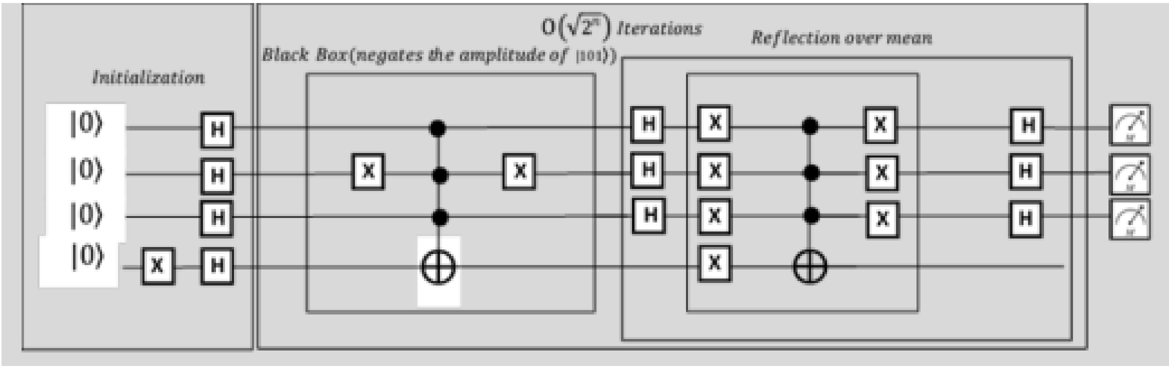
Similarly, the X gate rotates about the x-axis by 180°, rotating the angle θ e.g. X|0> = |1>, X|1> = |0>.

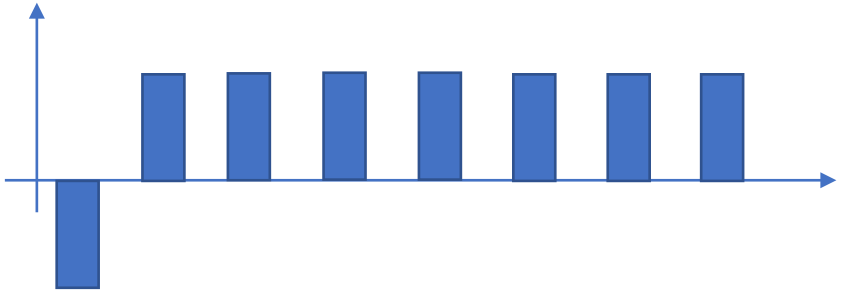


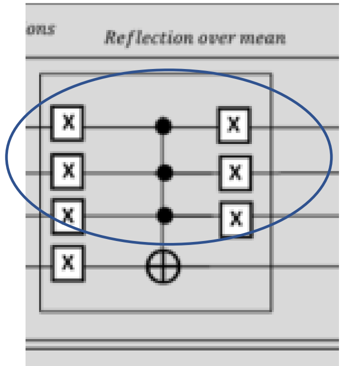
$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

We have seen in page 18 the two matrices for changing φ and θ in arbitraty amounts. They form a universal gate set – they can put a state anywhere on the Bloch Sphere. The gates Z and X are special cases of them.



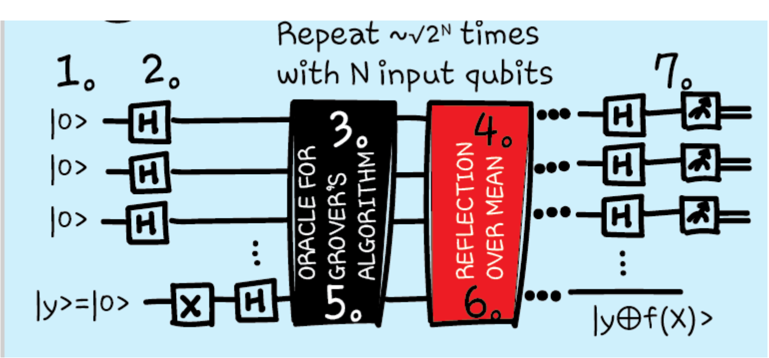


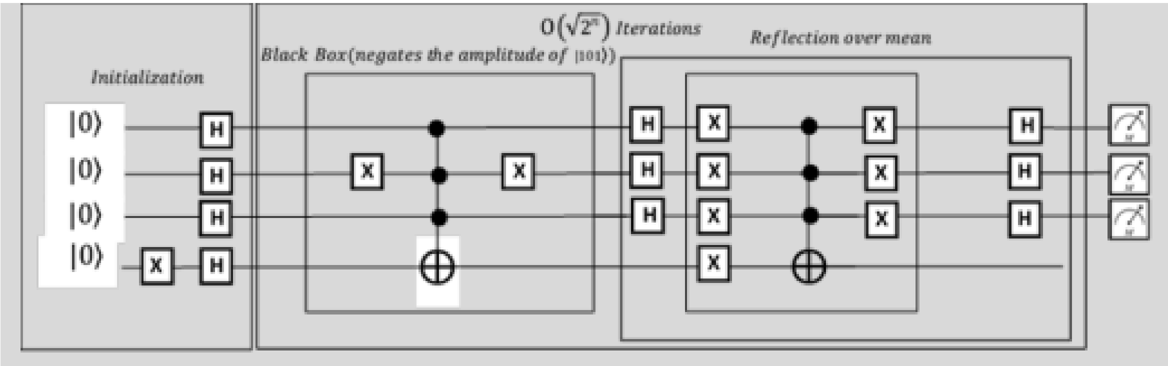


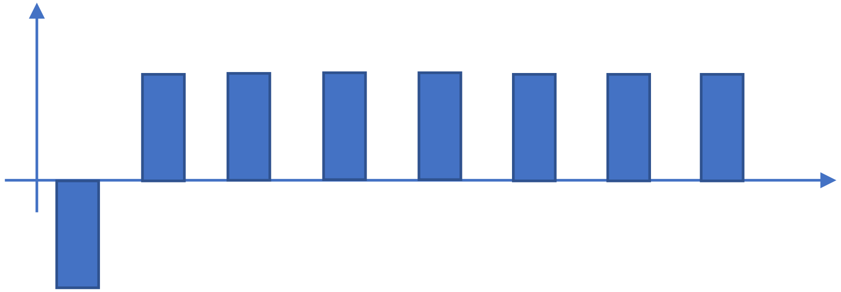


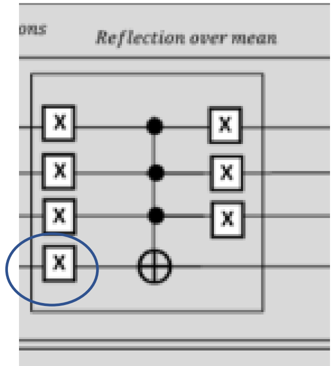
$$-a_0\,|000\rangle \otimes \left(-\frac{|0\rangle}{\sqrt{2}}+\frac{|1\rangle}{\sqrt{2}}\right) - a_1|001\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right) - a_2|010\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right) - a_3|011\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right) - a_4|100\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right) - a_5|101\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right) - a_6|110\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right) - a_7|111\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right)$$

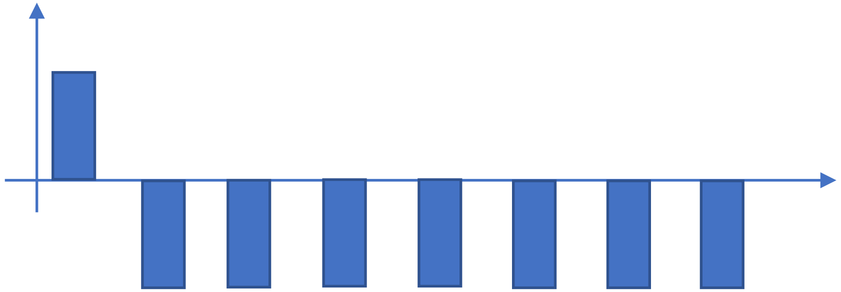
$$= \left(a_0|000\rangle - a_1|001\rangle - a_2|010\rangle - a_3|011\rangle - a_4|100\rangle - a_5|101\rangle - a_6|110\rangle - a_7|111\rangle\right) \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right)$$





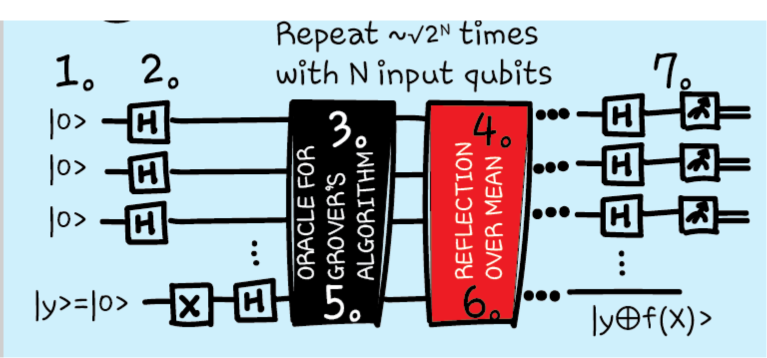


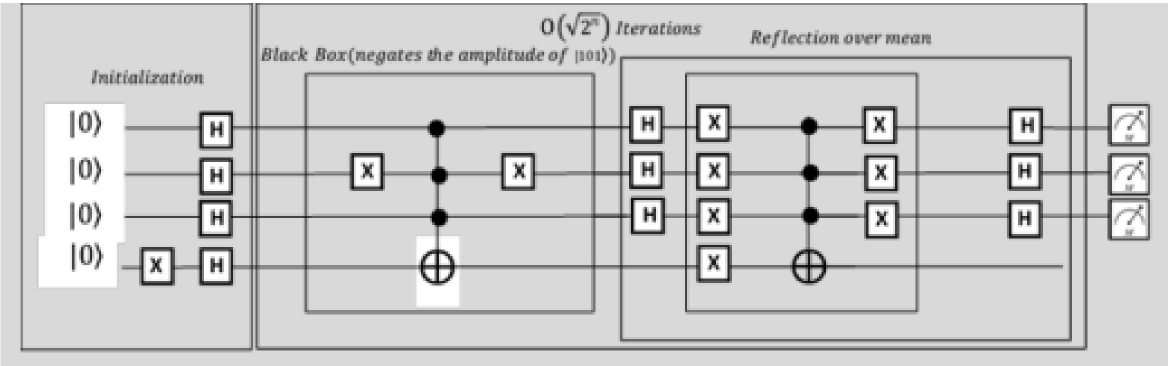


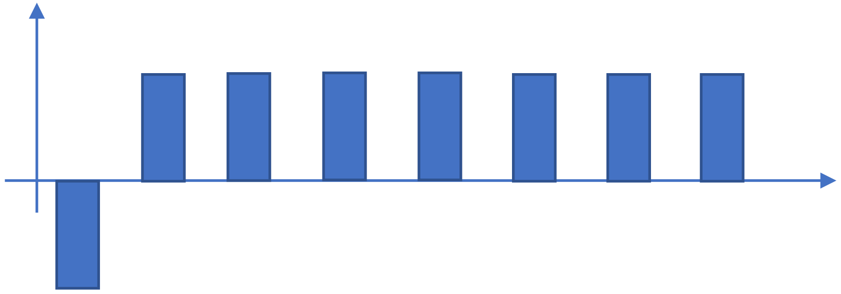


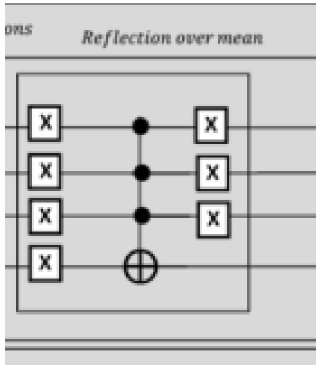
$$-a_0\,|000\rangle \otimes \left(-\frac{|0\rangle}{\sqrt{2}} + \frac{|1\rangle}{\sqrt{2}}\right) - a_1|001\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}} - \frac{|1\rangle}{\sqrt{2}}\right) - a_2|010\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}} - \frac{|1\rangle}{\sqrt{2}}\right) - a_3|011\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}} - \frac{|1\rangle}{\sqrt{2}}\right) - a_4|100\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}} - \frac{|1\rangle}{\sqrt{2}}\right) - a_5|101\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}} - \frac{|1\rangle}{\sqrt{2}}\right) - a_6|110\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}} - \frac{|1\rangle}{\sqrt{2}}\right) - a_7|111\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}} - \frac{|1\rangle}{\sqrt{2}}\right)$$

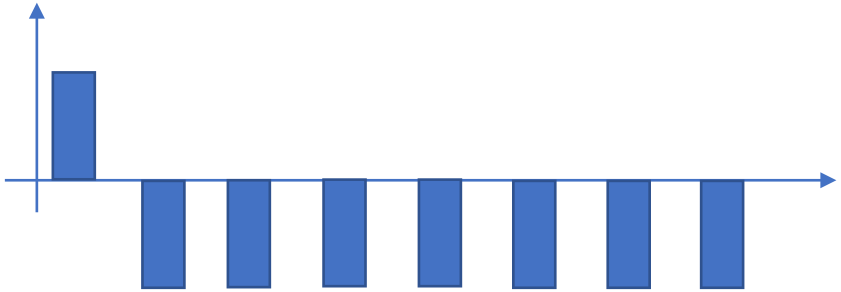
$$= (a_0|000\rangle - a_1|001\rangle - a_2|010\rangle - a_3|011\rangle - a_4|100\rangle - a_5|101\rangle - a_6|110\rangle - a_7|111\rangle) \otimes \left(\frac{|0\rangle}{\sqrt{2}} - \frac{|1\rangle}{\sqrt{2}}\right)$$





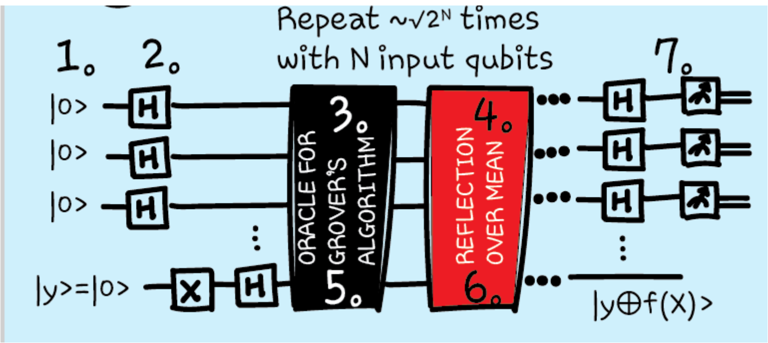


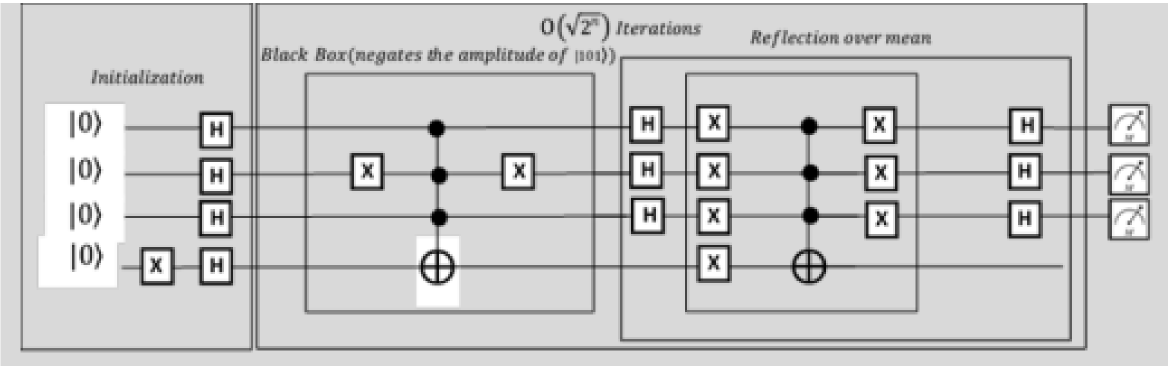


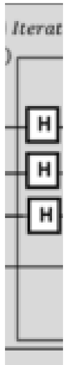


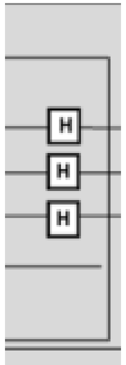
$$-a_0\,|000\rangle \otimes \left(-\frac{|0\rangle}{\sqrt{2}}+\frac{|1\rangle}{\sqrt{2}}\right) - a_1|001\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right) - a_2|010\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right) - a_3|011\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right) - a_4|100\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right) - a_5|101\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right) - a_6|110\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right) - a_7|111\rangle \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right)$$

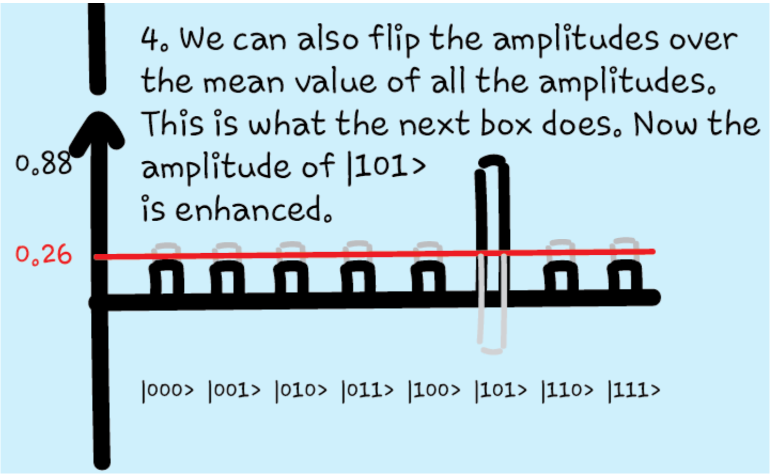
$$= (a_0|000\rangle - a_1|001\rangle - a_2|010\rangle - a_3|011\rangle - a_4|100\rangle - a_5|101\rangle - a_6|110\rangle - a_7|111\rangle) \otimes \left(\frac{|0\rangle}{\sqrt{2}}-\frac{|1\rangle}{\sqrt{2}}\right)$$









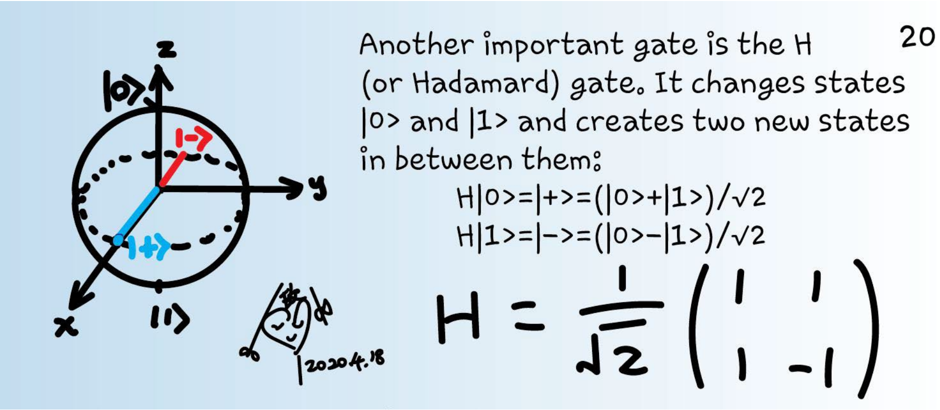


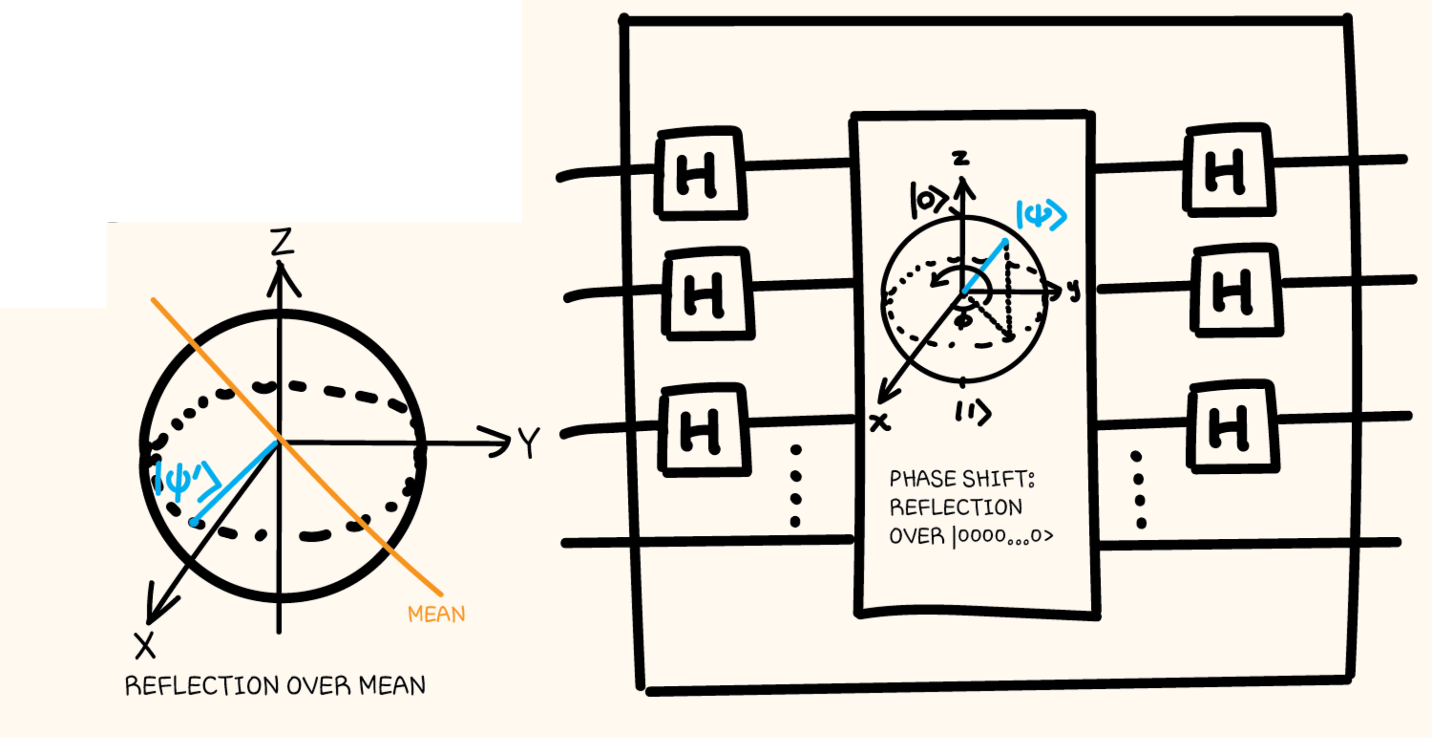
x_1 (original value)

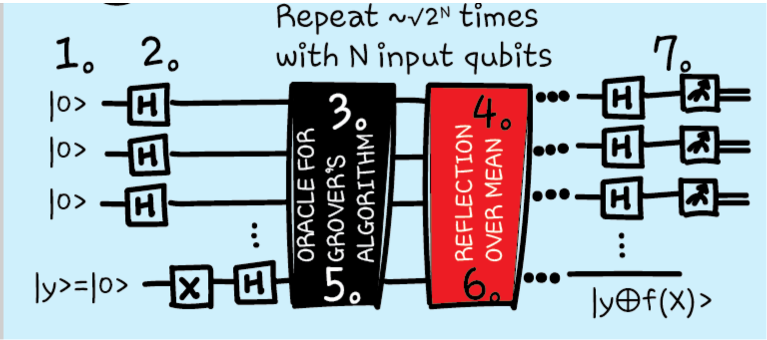
$mean = (x_1 + x_2)/2$

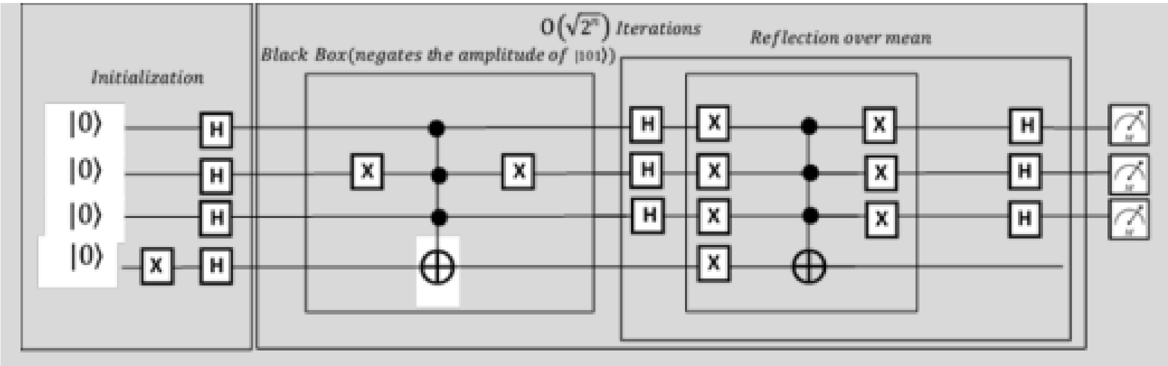
$\rightarrow x_2 = 2 * mean - x_1$

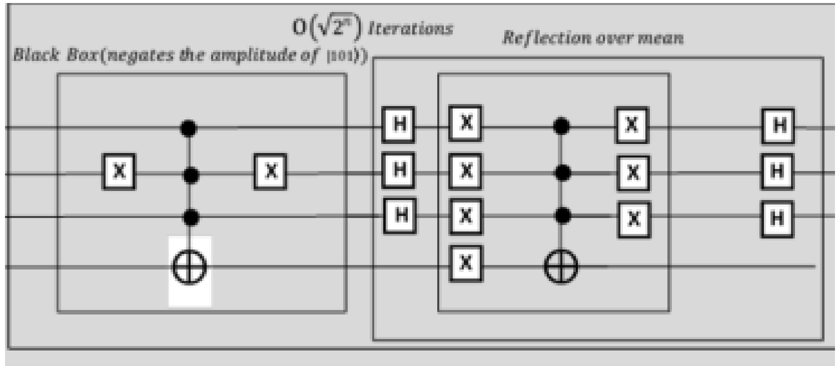
x_2 (new value after reflection over mean)

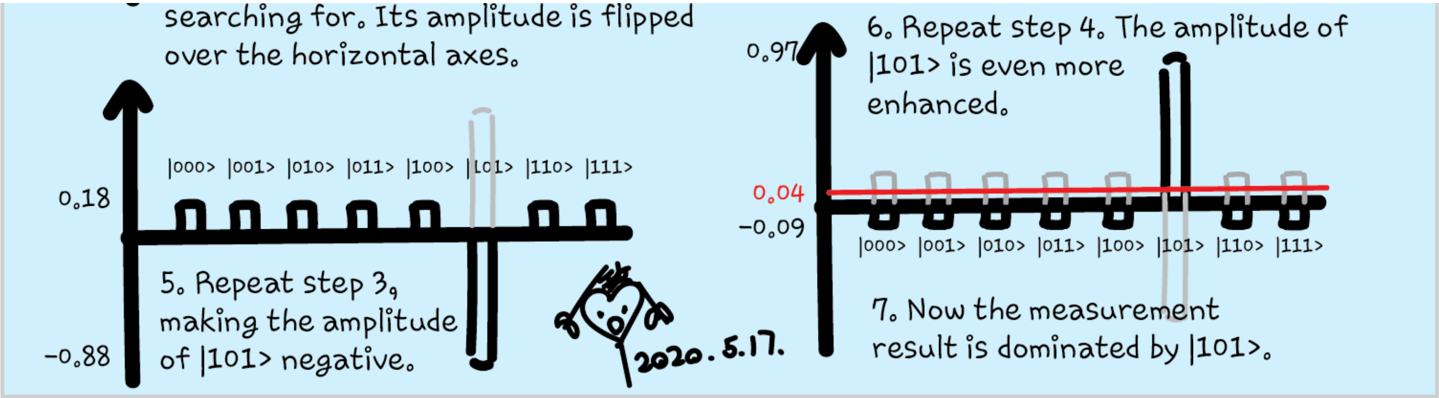



Z
Y
X
|ψ'⟩
MEAN
REFLECTION OVER MEAN
H
z
|0⟩
|ψ⟩
y
x
|1⟩
PHASE SHIFT:
REFLECTION
OVER |0000...0⟩

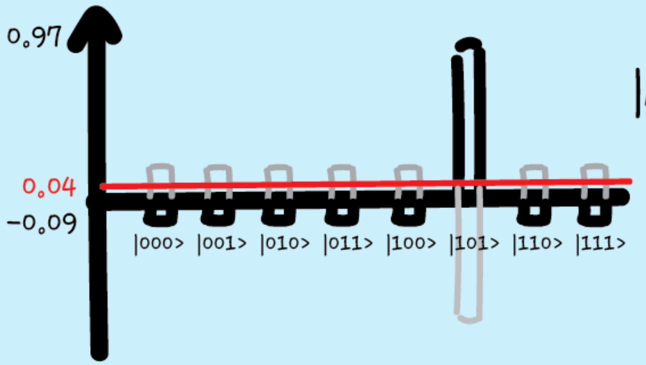

Repeat ~√2^N times
with N input qubits
1.
2.
7.
|0>
|0>
|0>
|y>=|0>
3.
ORACLE FOR GROVER'S ALGORITHM
5.
4.
REFLECTION OVER MEAN
6.
|y⊕f(X)>

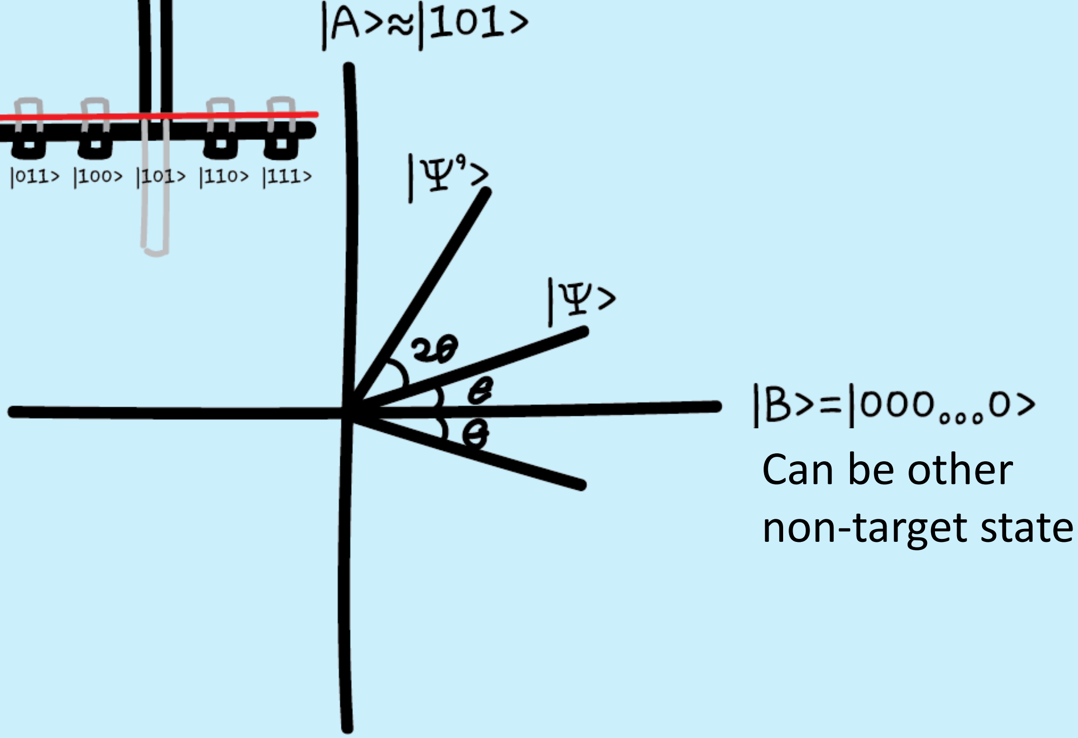

O(√2^n) Iterations
Black Box(negates the amplitude of |101⟩)
Reflection over mean
Initialization
|0⟩
|0⟩
|0⟩
|0⟩


O(√2^n) Iterations
Black Box(negates the amplitude of |101⟩)
Reflection over mean

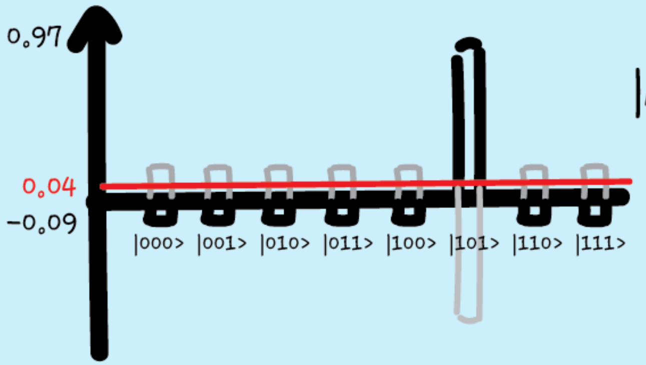

searching for. Its amplitude is flipped
over the horizontal axes.
0.18
-0.88
|000> |001> |010> |011> |100> |101> |110> |111>
5. Repeat step 3,
making the amplitude
of |101> negative.
2020.5.17.
0.97
6. Repeat step 4. The amplitude of
|101> is even more
enhanced.
0.04
-0.09
|000> |001> |010> |011> |100> |101> |110> |111>
7. Now the measurement
result is dominated by |101>.

How many iterations in Grover's algorithm is enough?



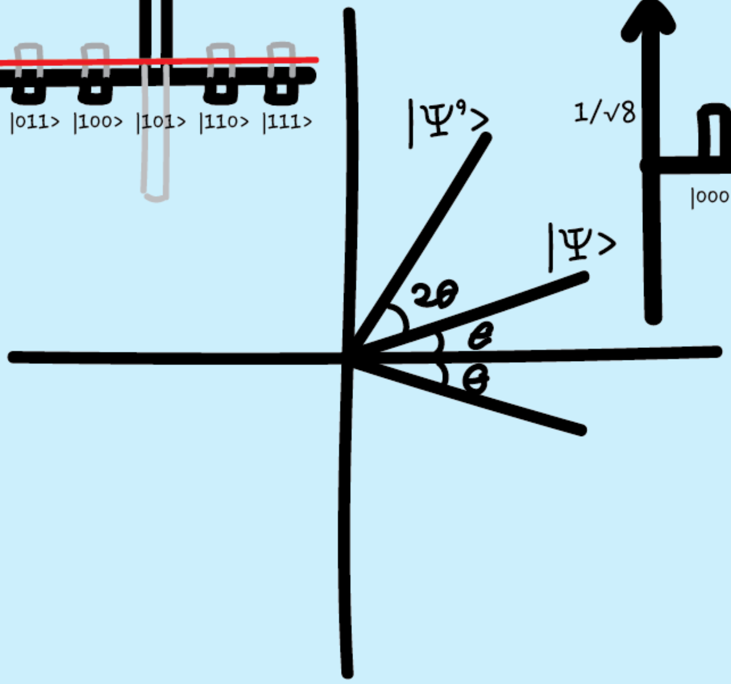


How many iterations in Grover's algorithm is enough?

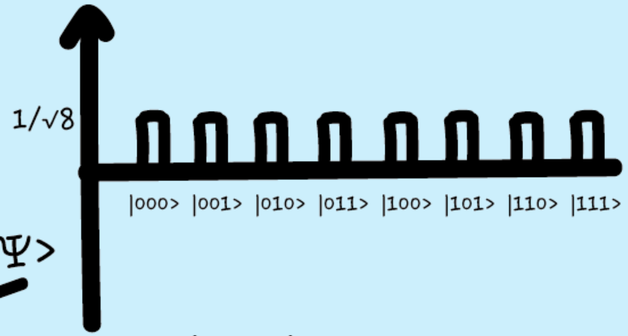


H gates make $|\Psi\rangle = \cos\theta|B\rangle + \sin\theta|A\rangle$
with $\sin\theta = 1/\sqrt{2^N}$

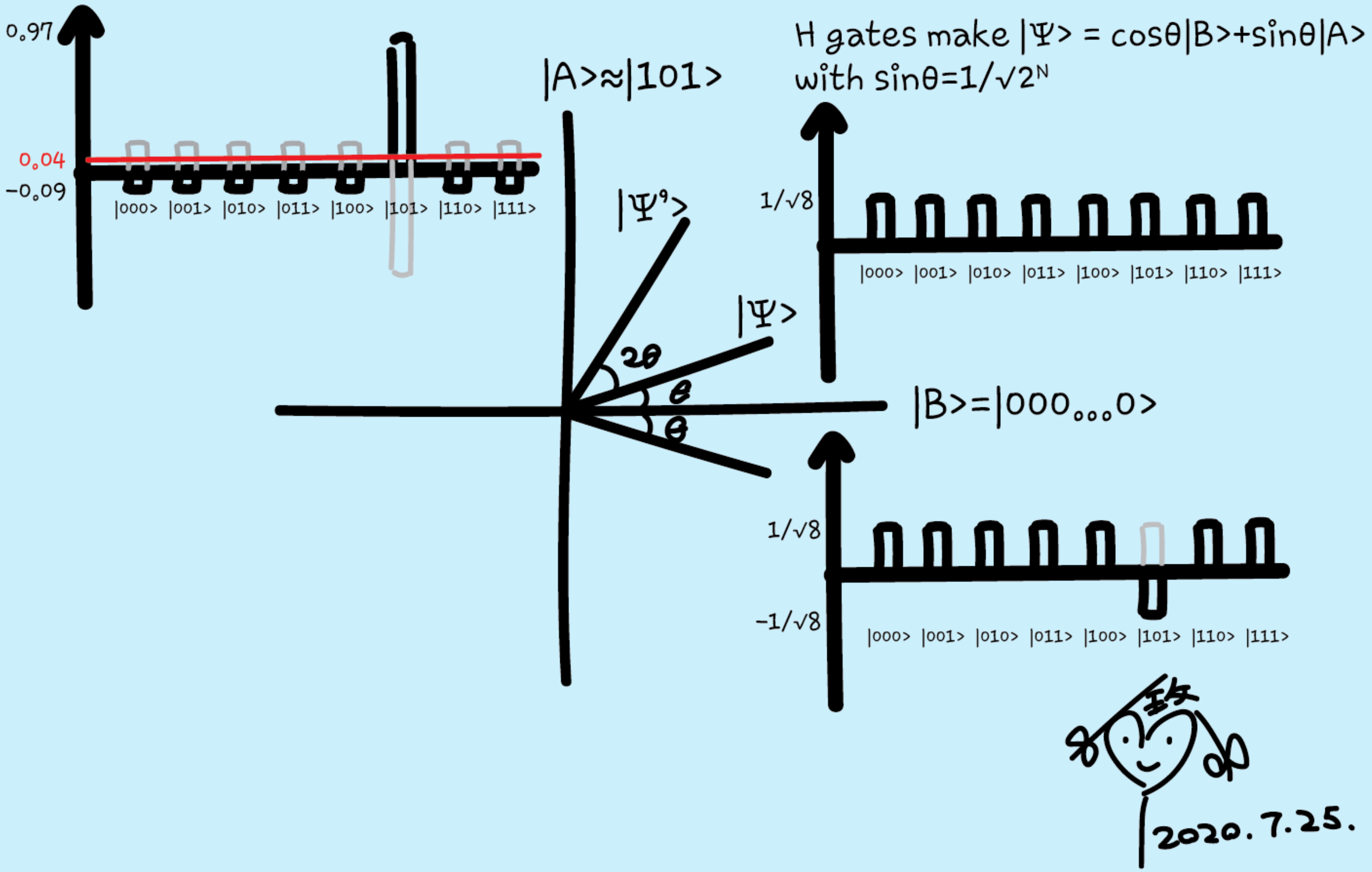
$|A\rangle \approx |101\rangle$



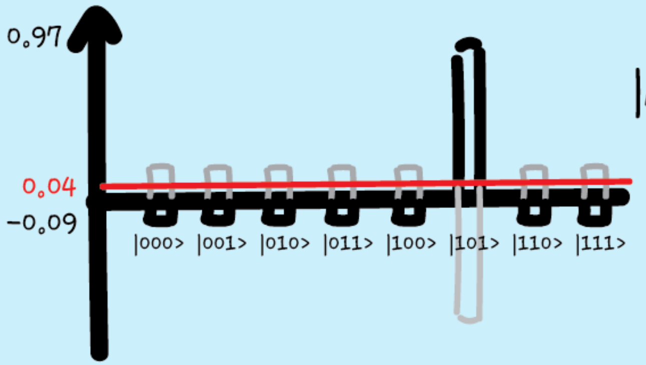
$|B\rangle = |000...0\rangle$



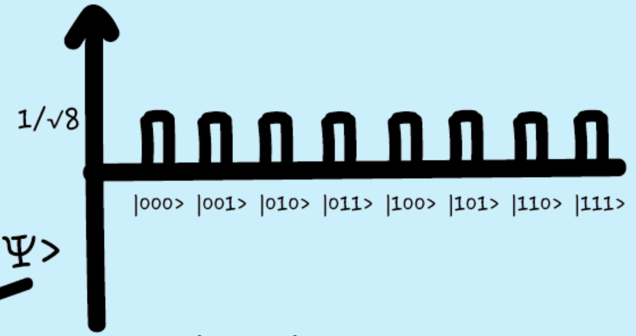
How many iterations in Grover's algorithm is enough?

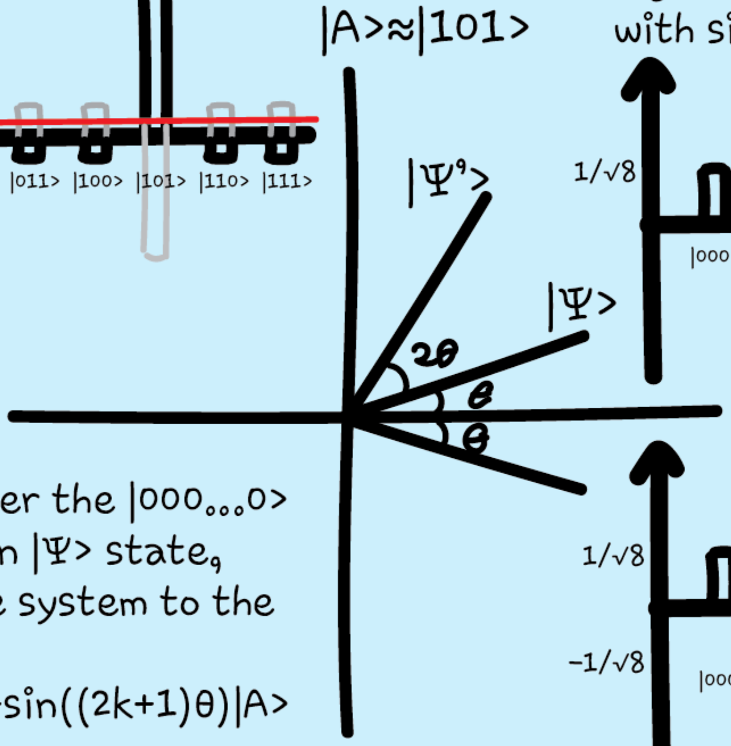


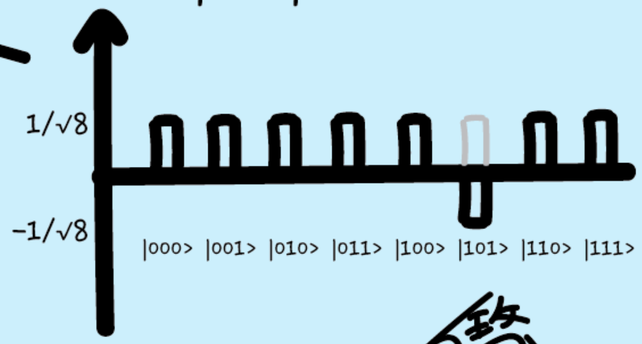
How many iterations in Grover's algorithm is enough?



H gates make $|\Psi\rangle = \cos\theta|B\rangle + \sin\theta|A\rangle$
with $\sin\theta = 1/\sqrt{2^N}$

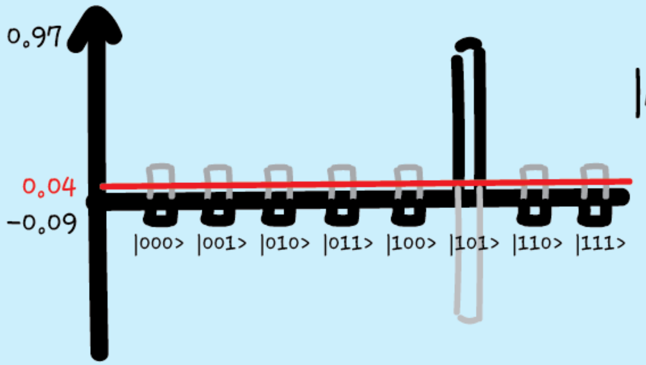


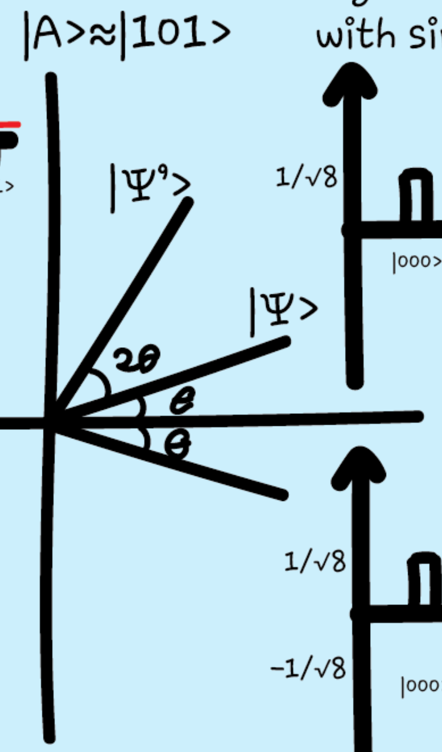




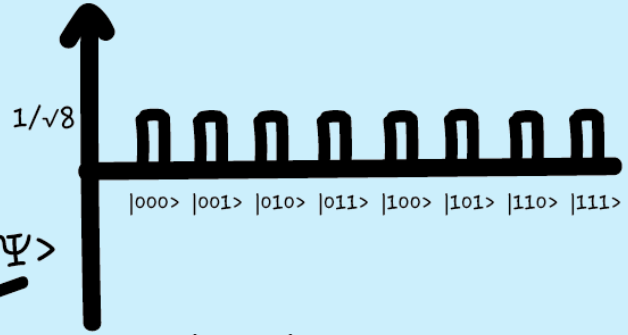
Keep reflecting over the $|000...0\rangle$ state and the mean $|\Psi\rangle$ state, k times, brings the system to the new state: $|\Psi'\rangle$
$= \cos((2k+1)\theta)|B\rangle + \sin((2k+1)\theta)|A\rangle$

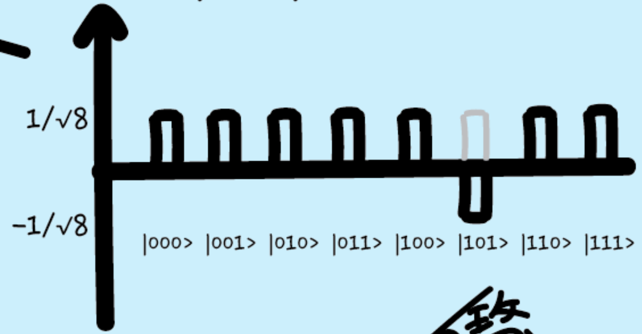
How many iterations in Grover's algorithm is enough?





H gates make $|\Psi\rangle = \cos\theta|B\rangle + \sin\theta|A\rangle$ with $\sin\theta = 1/\sqrt{2^N}$





Keep reflecting over the $|000...0\rangle$ state and the mean $|\Psi\rangle$ state, k times, brings the system to the new state : $|\Psi'\rangle$
$= \cos((2k+1)\theta)|B\rangle + \sin((2k+1)\theta)|A\rangle$

It is closest to $|A\rangle$ when:
$\sin((2k+1)\theta) \approx 1 \Rightarrow (2k+1)\theta \approx \pi/2 \Rightarrow k \approx \pi/(4\theta) - 1/2$
That's why the number of iterations is on the order of $\sqrt{(2^N)}$.

2020.7.25.

Quantum katas

Set up Grover's algorithm from scratch	https://github.com/microsoft/QuantumKatas/tree/master/GroversAlgorithm
Use Grover's algorithm	https://github.com/microsoft/QuantumKatas/tree/master/tutorials/ExploringGroversAlgorithm
Visualize Grover's algorithm	https://github.com/microsoft/QuantumKatas/tree/master/GraphColoring
Decorating the Christmas tree using Grover's search	https://github.com/tcNickolas/MiscQSharp/tree/master/DecoratingTheTree

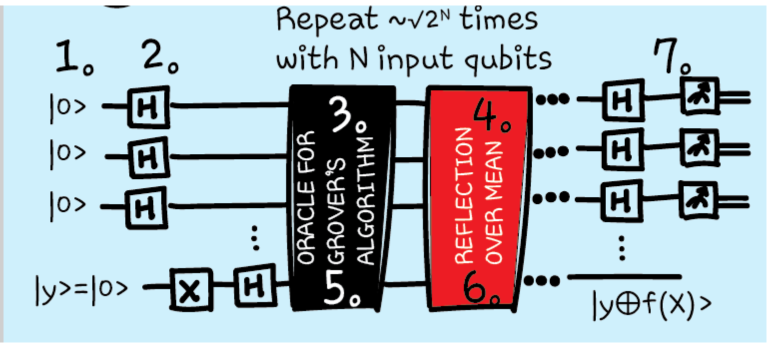
Q# exercise:

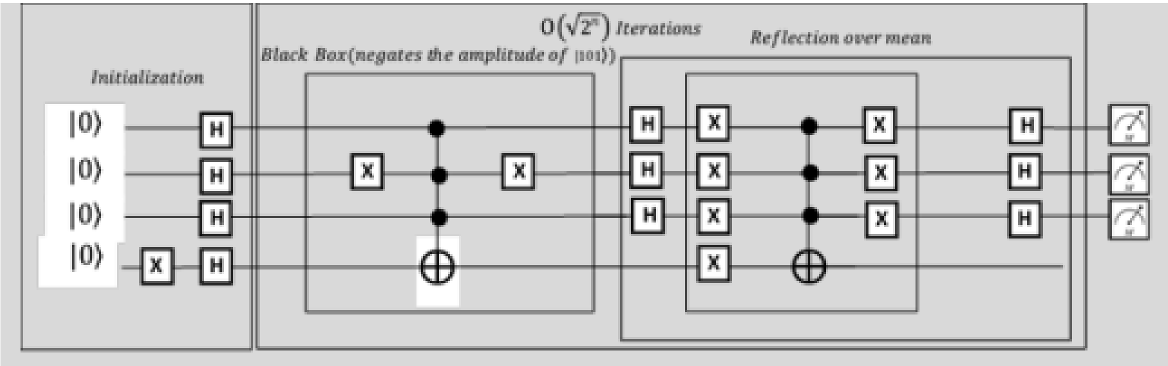
Quantum Katas

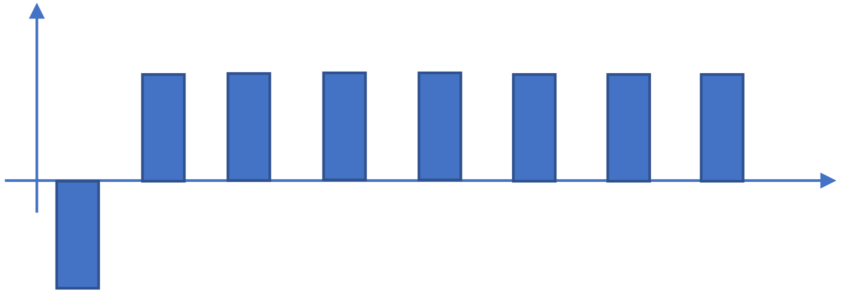
https://github.com/Microsoft/QuantumKatas

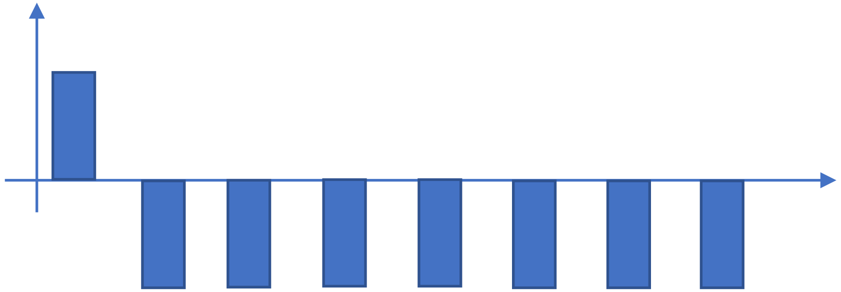
- **GroversAlgorithm**
  - Task 1.1, 2.1-2.3
- Grover’s tutorial on Quantum Development Kit

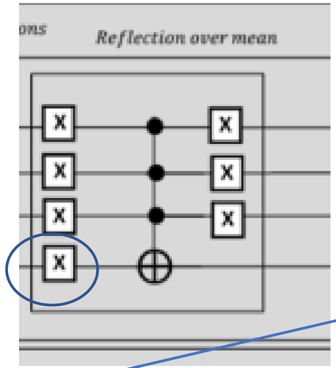
http://docs.microsoft.com/quantum

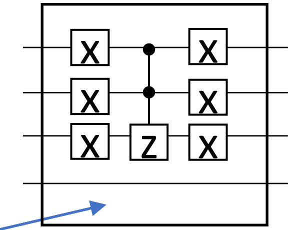




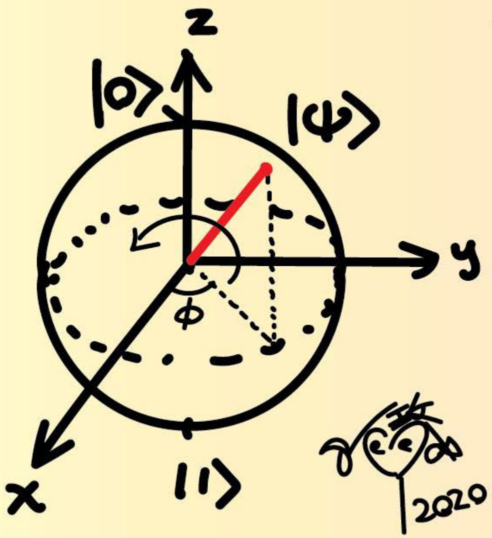






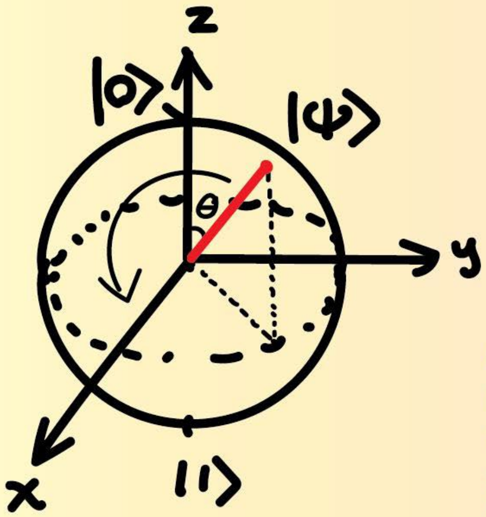


Introduce the "-" sign



To change the phase φ, we have a commonly used gate, Z, which rotates about the z-axis by 180°.

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$



Similarly, the X gate rotates about the x-axis by 180°, rotating the angle θ e.g. X|0> = |1>, X|1> = |0>.

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

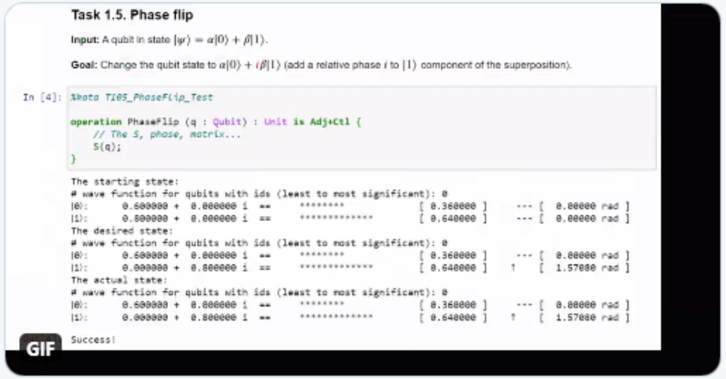
We have seen in page 18 the two matrices for changing φ and θ in arbitraty amounts. They form a universal gate set – they can put a state anywhere on the Bloch Sphere. The gates Z and X are special cases of them.

For certificate 1

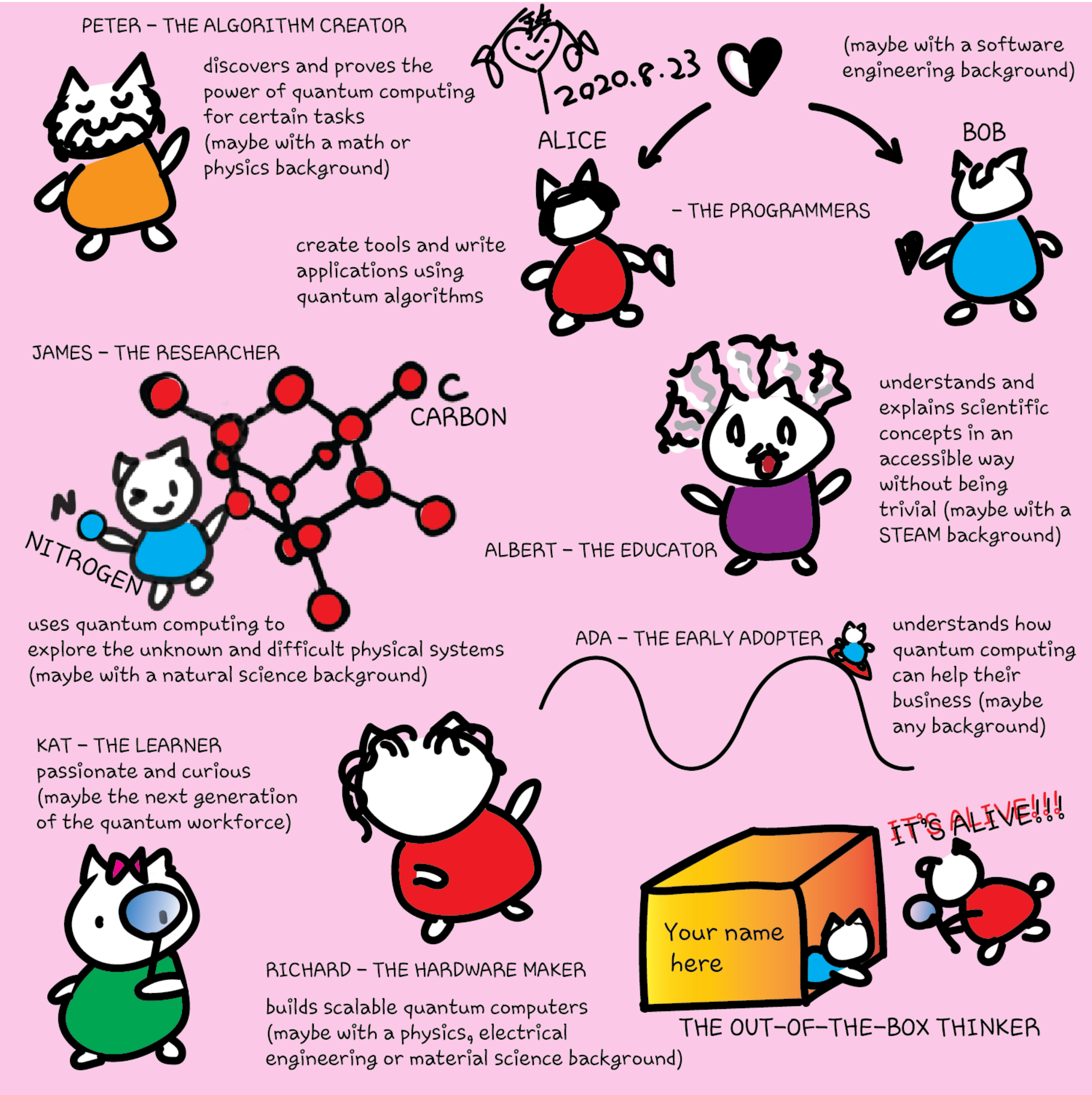
- Complete any one quantum katas
- Take a screenshot or photo
- Post on Twitter or LinkedIn
- Tag the following
- **Twitter:** @KittyArtPhysics @MSFTQuantum @QSharpCommunity #QSharp #QuantumComputing #comics #physics
- **LinkedIn:** @Kitty Y. M Yeung #MSFTQuantum #QSharp #QuantumComputing #comics #physics

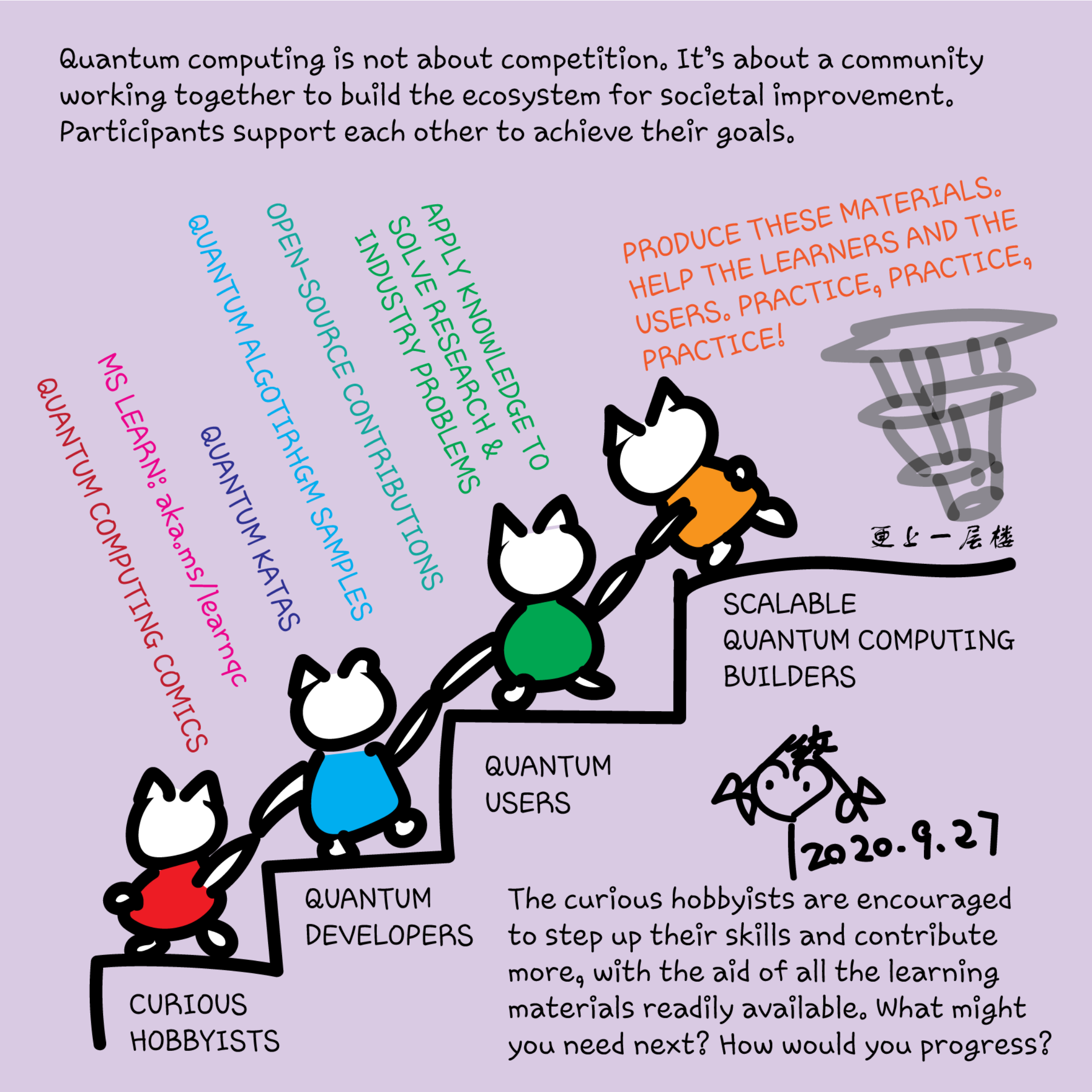
Roberto Aviles A. @rlaviles · May 7
Twitter: @KittyArtPhysics @MSFTQuantum @QSharpCommunity #QSharp #QuantumComputing #comics #physics

Katas Basic Gates, done (15 exercises.)



Tom @hb9xar · 2h
Thank you @KittyArtPhysics for teaching us some #QuantumComputing on @hackadayio and designing the #hackaday Schrödingers cat.
Looks *very* nice on a mug.


PETER – THE ALGORITHM CREATOR
discovers and proves the power of quantum computing for certain tasks (maybe with a math or physics background)
2020.8.23
ALICE
BOB
– THE PROGRAMMERS
(maybe with a software engineering background)
create tools and write applications using quantum algorithms
JAMES – THE RESEARCHER
C
CARBON
N
NITROGEN
uses quantum computing to explore the unknown and difficult physical systems (maybe with a natural science background)
ALBERT – THE EDUCATOR
understands and explains scientific concepts in an accessible way without being trivial (maybe with a STEAM background)
ADA – THE EARLY ADOPTER
understands how quantum computing can help their business (maybe any background)
KAT – THE LEARNER
passionate and curious (maybe the next generation of the quantum workforce)
RICHARD – THE HARDWARE MAKER
builds scalable quantum computers (maybe with a physics, electrical engineering or material science background)
Your name here
IT'S ALIVE!!!
THE OUT-OF-THE-BOX THINKER

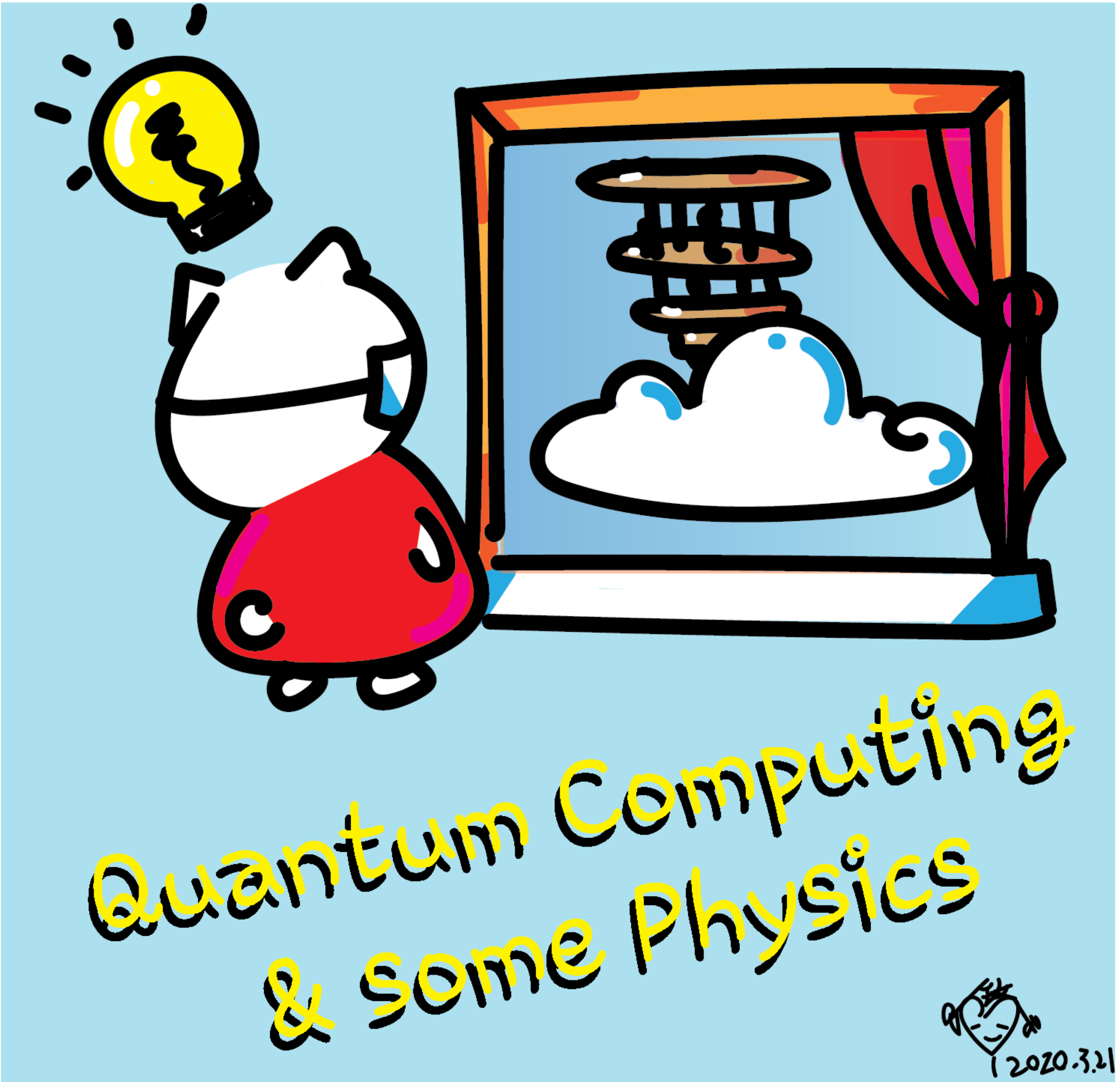

Quantum computing is not about competition. It's about a community working together to build the ecosystem for societal improvement. Participants support each other to achieve their goals.
QUANTUM COMPUTING COMICS
MS LEARN: aka.ms/learnqc
QUANTUM KATAS
QUANTUM ALGOTIRHGM SAMPLES
OPEN-SOURCE CONTRIBUTIONS
APPLY KNOWLEDGE TO SOLVE RESEARCH & INDUSTRY PROBLEMS
PRODUCE THESE MATERIALS. HELP THE LEARNERS AND THE USERS. PRACTICE, PRACTICE, PRACTICE!
更上一层楼
SCALABLE QUANTUM COMPUTING BUILDERS
QUANTUM USERS
QUANTUM DEVELOPERS
CURIOUS HOBBYISTS
2020.9.27
The curious hobbyists are encouraged to step up their skills and contribute more, with the aid of all the learning materials readily available. What might you need next? How would you progress?

Questions

- Post in chat or on Hackaday project
  https://hackaday.io/project/168554-quantum-computing-through-comics
- FAQ: Past Recordings on Hackaday project or my YouTube https://www.youtube.com/c/DrKittyYeung

Class structure

- Comics on Hackaday – Quantum Computing through Comics every Sun
- 30 mins – 1 hour every Sun, one concept (theory, hardware, programming), Q&A
- Contribute to Q# documentation http://docs.microsoft.com/quantum
- Coding through Quantum Katas https://github.com/Microsoft/QuantumKatas/
- Discuss in Hackaday project comments throughout the week
- Take notes

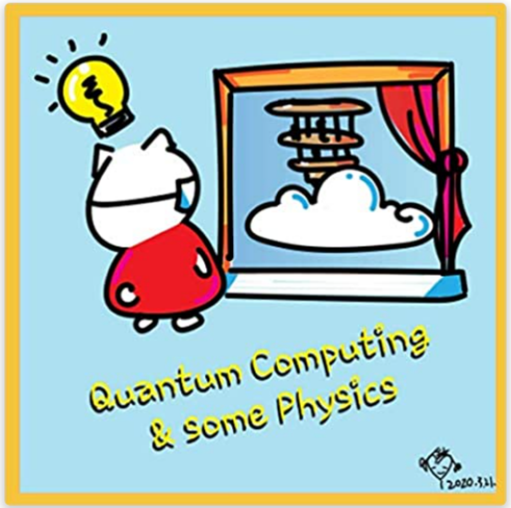


ASIN : B08HGLPZXP in 13 markets

Books | Advanced Search | New Releases | Best Sellers & More | Children's Books | Textbooks | Textbook Rentals | Magazines | Best Books of the Month

Books › Comics & Graphic Novels › Graphic Novels

Look inside



Quantum Computing & Some Physics: The Quantum Computing Comics Notebook

Paperback – September 3, 2020

by Dr. Kitty Yeung (Author)

See all formats and editions

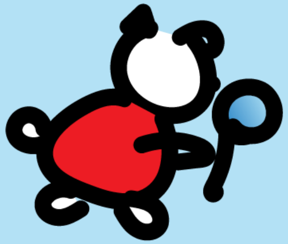
Paperback
$19.98

1 New from $19.98

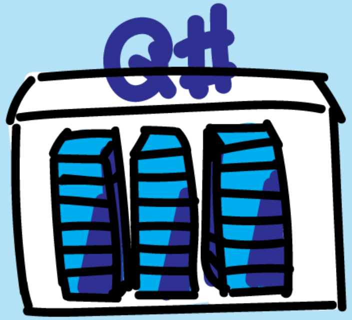
Learn about quantum computing through an intuitive series of comics. It is both a book and a notebook, in which readers can note down their thoughts on the back of the comics. The book provides a high-level guide to the basic concepts of quantum computing, linear algebra, and quantum algorithms. Commonly used quantum hardware architectures are also described in the comics. Learners at any age with any background can get something out of this comics. The

THE SUNDAY SPECIALS

September 13
Prof. Terrill Frantz
Quantum Cryptography



~~September 20~~ October 25
Prof. Chris Ferrie
Quantum Tomography

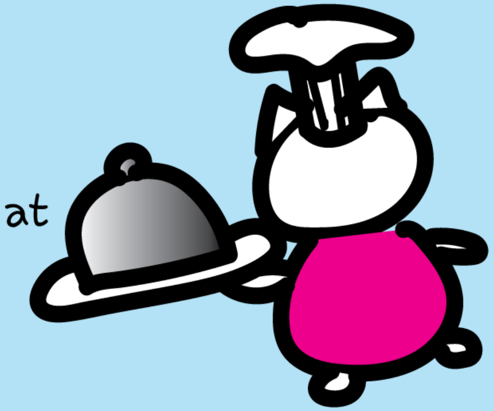


September 27
Rolf Huisman
Introducing the open source
Q# Community project qTRIL

October 11
Dr. Maria Schuld
Quantum Machine Learning

October 18
Dr. Michael Beverland
Quantum Error Correction

October 3
Kitty speaking at
Zen4Makers



2020.9.13.

aka.ms/learnqc

3800 XP

Quantum computing foundations

47 min remaining • Learning Path • 2 of 4 modules completed

Beginner Developer Quantum Development Kit Quantum

Intrigued by quantum computing but don't know where to start? This learning path helps prepare you for this exciting next generation of computing.

After completing this learning path, you'll be able to:

- Explain the fundamental concepts of quantum computing.
- Build basic quantum programs by using the Quantum Development Kit and Q#.
- Identify the kinds of problems quantum algorithms can solve more efficiently than classical algorithms.

Prerequisites

None

Continue > Bookmark Add to collection

Contribute and demonstrate your skills

- **microsoft/Quantum**: Samples and tools to help get started with the Quantum Development Kit.
- **microsoft/QuantumLibraries**: Standard and domain-specific libraries for the Quantum Development Kit.
- **microsoft/QuantumKatas**: Self-paced programming exercises for learning quantum computing and the Q# programming language.
- **microsoft/qsharp-compiler**: The Q# compiler, Visual Studio extension, and Visual Studio Code extension.
- **microsoft/qsharp-runtime**: Simulation framework, code generation, and simulation target machines for the Quantum Development Kit.
- **microsoft/iqsharp**: Jupyter kernel and Python host functionality for Q#, as well as Docker images for using IQ# in cloud environments.
- **MicrosoftDocs/quantum-docs-pr**: Source code for the documentation published at https://docs.microsoft.com/quantum.

microsoft.com/en-us/quantum/development-kit

Experience the impact of quantum solutions today

We're bringing quantum apps to life with an easy to use tool set, deep integration with leading development environments, and open-source resources.

The Microsoft Quantum Development Kit is the fastest path to quantum development.

Watch now

Learn to set up the QDK >

Help us create new quantum learning content for people like you

Take our survey >

Hacktoberfest
in Microsoft GitHub repos

Hacktoberfest® is open to everyone in our global community.

- Hacktoberfest is a celebration open to everyone in our global community.
- Pull requests can be made in any GitHub-hosted repositories/projects.
- You can sign up anytime between October 1 and October 31.

To earn your Hacktoberfest tee or tree reward, you must register and make four valid pull requests (PRs) between October 1-31 (in any time zone). PRs can be made to any public repo on GitHub, not only the ones with issues labeled Hacktoberfest. If a maintainer reports your pull request as spam or behavior not in line with the project's code of conduct, you will be ineligible to participate. This year, the first 70,000 participants who successfully complete the challenge will be eligible to receive a prize.

For more participation details: https://hacktoberfest.digitalocean.com/
Quantum Hacktoberfest blog: https://devblogs.microsoft.com/qsharp/celebrating-our-open-source-community-with-hacktoberfest/