

Quantum Tokyo

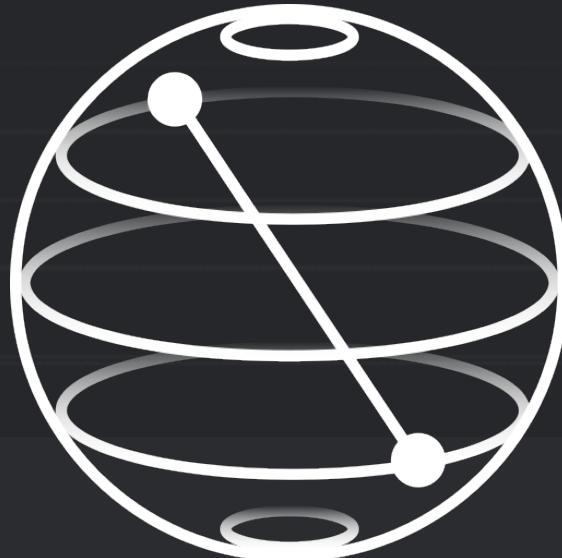
量子コンピューター入門ハンズオン

～グローバーのアルゴリズムまで～



沼田 祢史
Kifumi Numata, Qiskit Advocate

Mar 31, 2022



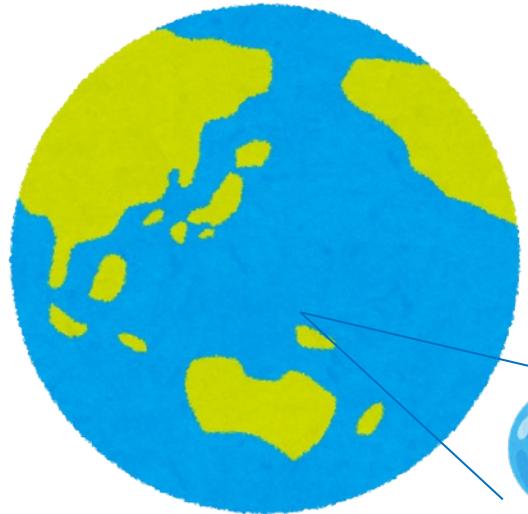
Qt

本日の内容

- 量子コンピューター入門
- IBM Quantum Composer ハンズオン
 - 基本の量子ゲートの操作
 - グローバーのアルゴリズムの実装
 - 実機量子コンピューターでの実験
- 事前準備：IBM Quantumへのログイン
(参照：『IBM Quantum Composerの登録（くわしいバージョン）』
<https://qiita.com/kifumi/items/7ac33ab7939d2dd796d0>)
- 参加対象：量子コンピューター初心者の方
- 今日の資料：
https://github.com/quantum-tokyo/qiskit-hands-on/composer/20220331_IQX_grover.pdf

量子とは？

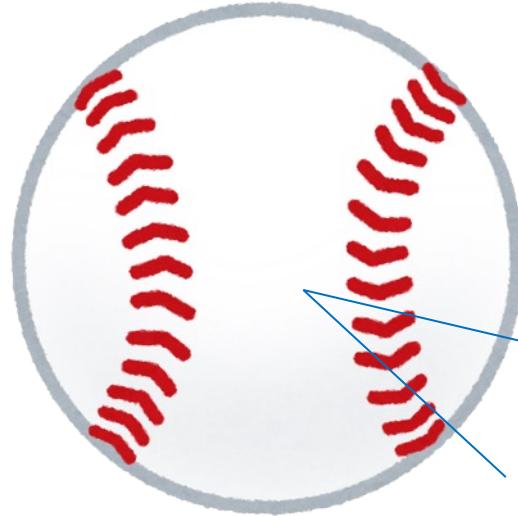
あらゆるものは、「原子」からできています。



直径1万3000km

同じ比率
↔

ビー玉
直径1cm



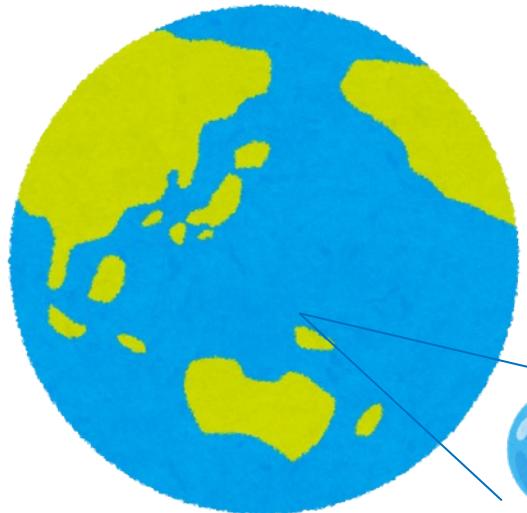
直径7cm



原子
直径0.1nm
ナノメートル

量子とは？

あらゆるものは、「原子」からできています。



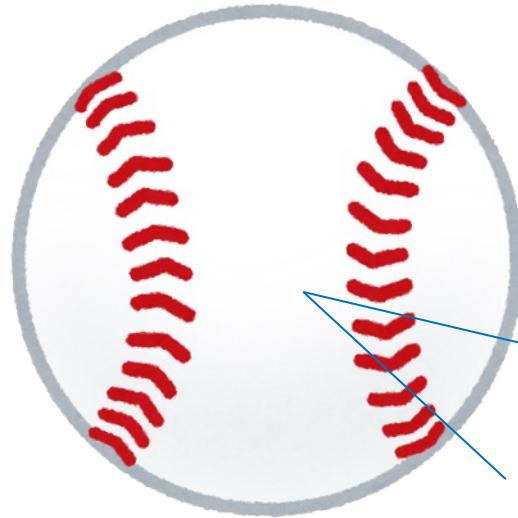
直径1万3000km

同じ比率

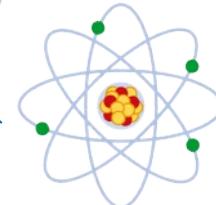


ビー玉

直径1cm



直径7cm



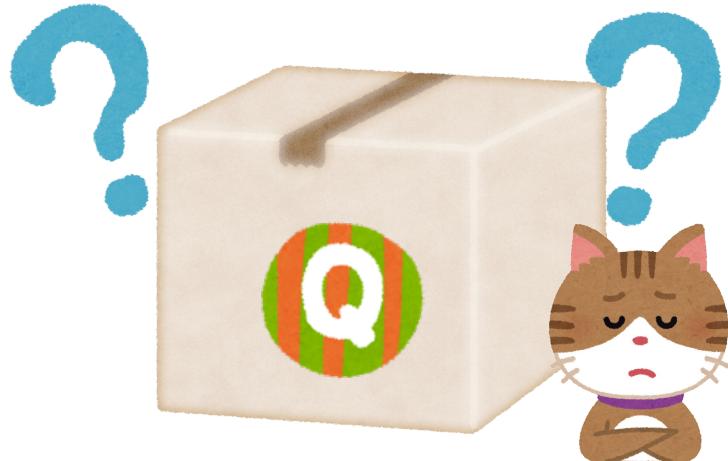
原子
直径0.1nm
ナノメートル

ミクロな世界

原子や原子より小さい物質を量子と呼びます。
このミクロな物質は、私たちの常識では説明できない、
不思議なふるまいをします。参考：Newtonライト2.0『量子論』

量子の不思議な現象：量子重ね合わせ

量子は1つしかなくとも、複数の状態を同時にとることができます。



量子の不思議な現象：量子重ね合わせ

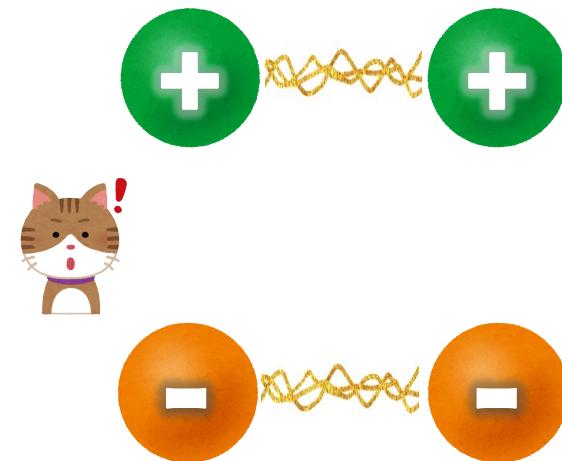
量子は1つしかなくとも、たくさんの状態を同時にとることができます。



例えば、+と-の2つの状態が共存していて、観測するとどちらかの状態に決まります。（分らなくて大丈夫です！量子とは直感では理解できないものです！）

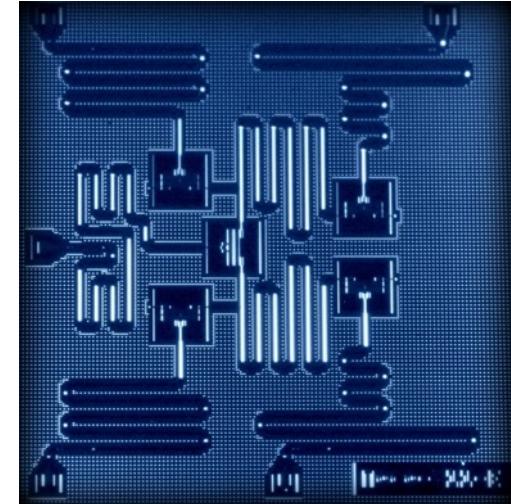
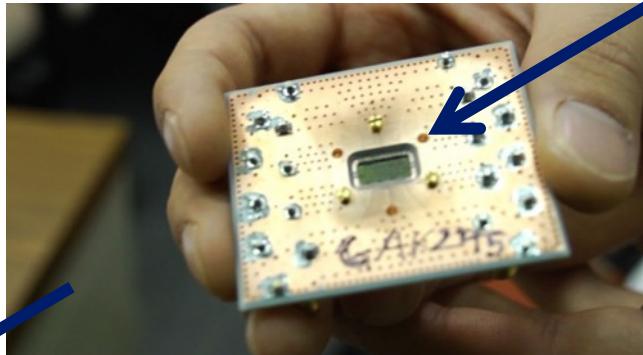
量子の不思議な現象：量子もつれ

「量子もつれ」という特別な関係のふたごの量子は、
片方の状態を測定すると、もう片方の状態も分かります。
(片方しか測定していないのに！)



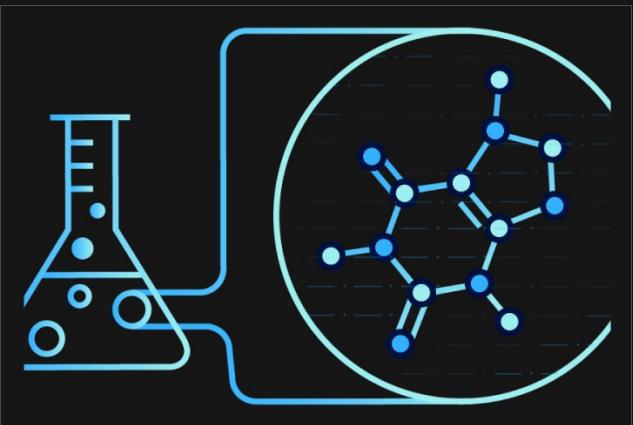
分らなくても大丈夫です！この量子の不思議な現象を使って量子計算を行います！

量子コンピューター

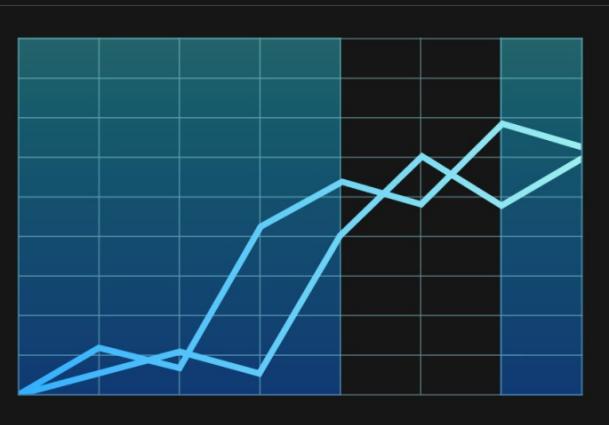


約-273°C (0.015K)の低温で量子状態を実現
ケルビン

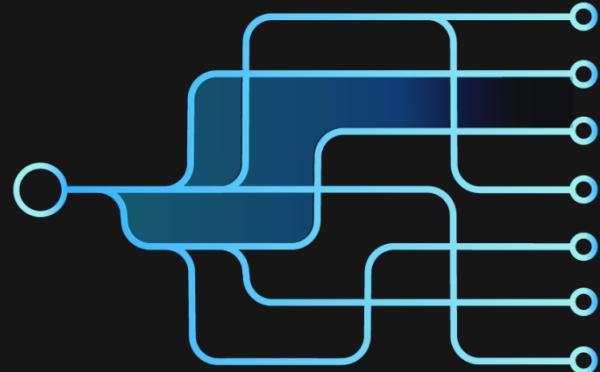
量子コンピューターの適用分野



量子化学

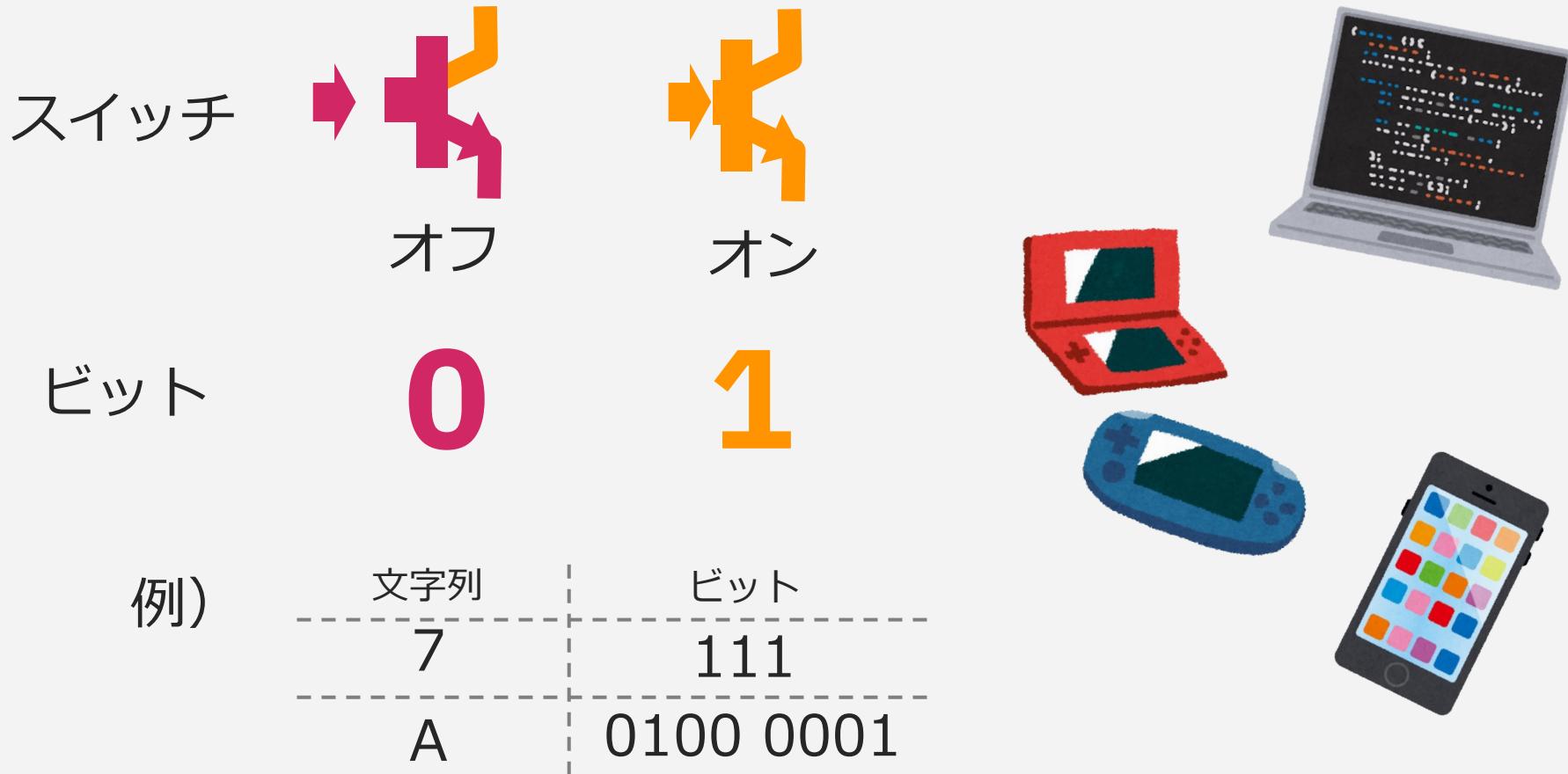


金融

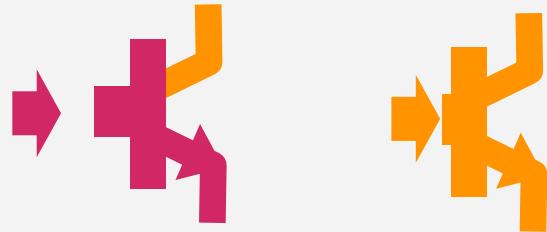


最適化/機械学習

コンピューターの中は、ビットで計算



いつも使っている コンピューターのビット



0 または 1

どちらか

いつも使っている
コンピューターのビット

0 または **1**

どちらか

量子コンピューターの
量子ビット

0 と **1**

両方

いつも使っている
コンピューターのビット

0 または **1**

どちらか

コイン

表

おもて



コイン

裏

うら

量子コンピューターの
量子ビット

0 と **1**

両方

いつも使っている
コンピューターのビット

0 または 1

どちらか

コイン

表

おもて



コイン

裏

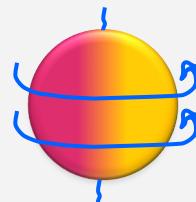
うら

量子コンピューターの
量子ビット

0 と 1

両方

くるくる回っているコイン（イメージ）



測定すると表か裏にバシッと決まる

いつも使っている
コンピューターのビット

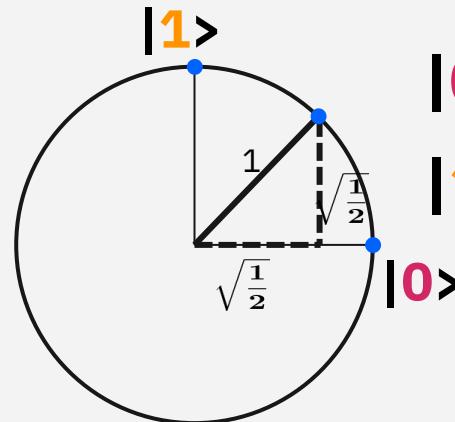
0 または 1

どちらか

量子コンピューターの
量子ビット

$\alpha \times |0\rangle + \beta \times |1\rangle$

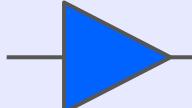
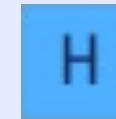
0 と 1 の「重ね合わせ」



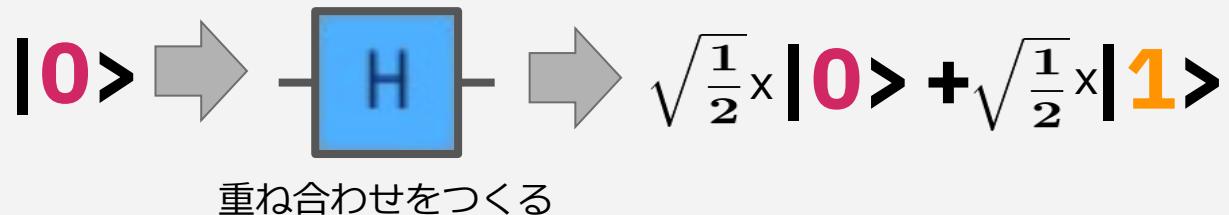
$$|0\rangle = 1 \times |0\rangle + 0 \times |1\rangle$$

$$|1\rangle = 0 \times |0\rangle + 1 \times |1\rangle$$

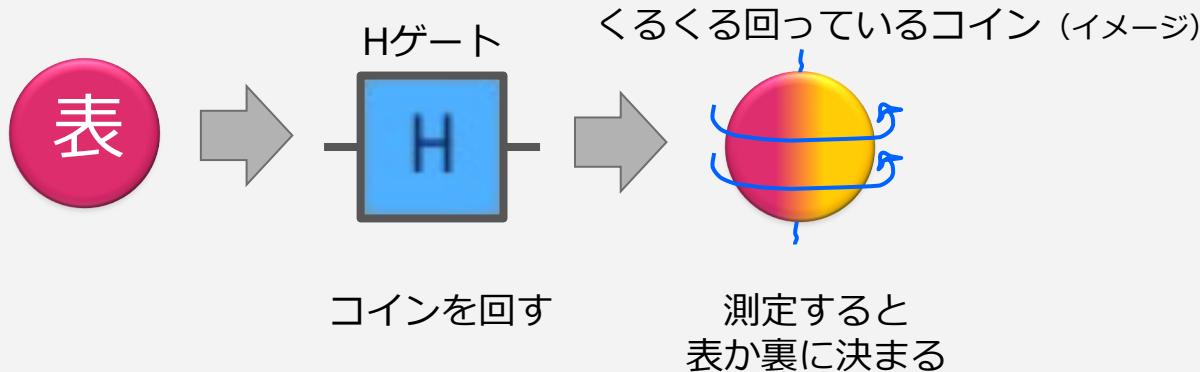
ビットと論理ゲート

	古典	量子
ビット	0 または 1	0と1の重ね合わせ $\alpha \times 0\rangle + \beta \times 1\rangle$
論理 ゲート	NOT, OR, AND など  	X, H, CNOT など   

量子コンピューターの計算方法



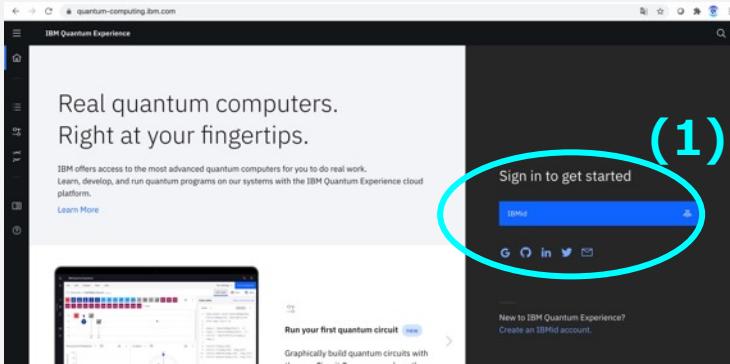
量子コンピューターの計算方法



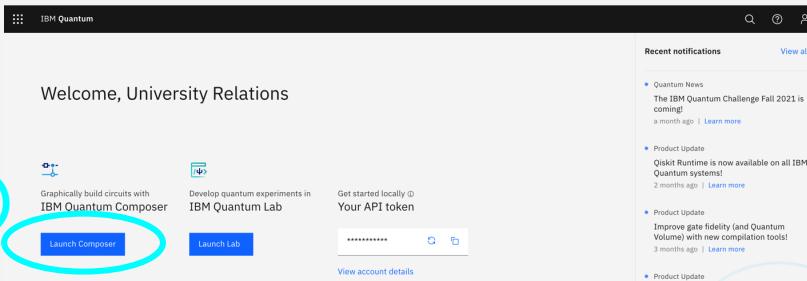
ハンズオン：いつしょにやってみましょう！

(1) IBM Quantum にログインします。URL:

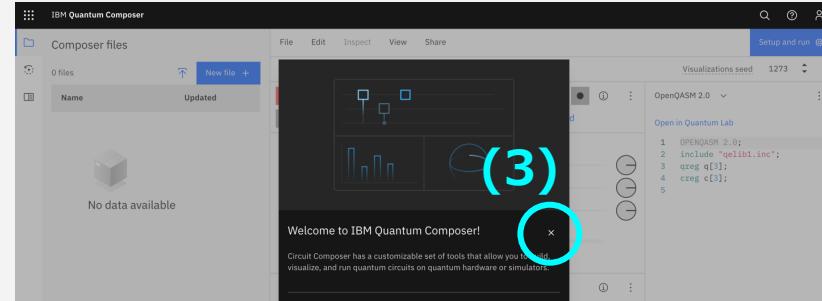
<https://quantum-computing.ibm.com/>



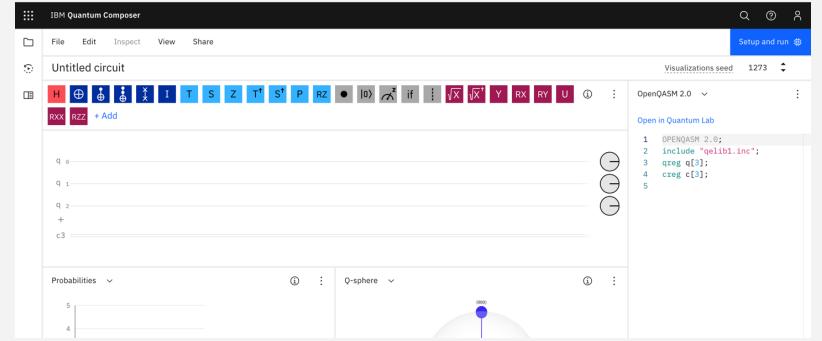
(2) 左の青アイコン「Launch Composer」をクリック。



(3) ポップアップウィンドウは右中程の「x」をクリックして、閉じます。



(4) この画面になつたら準備完了です。



1量子ビット回路

IBM Quantum Composer

File Edit Inspect View Share Setup and run

Untitled circuit Visualizations seed 1273

H \oplus \oplus_1 \oplus_2 \otimes I T S Z T^\dagger S^\dagger P RZ \bullet $|0\rangle$ α^z if $| \rangle$ \sqrt{X} \sqrt{X}^\dagger Y RX RY U \dots

RXX RZZ + Add OpenQASM 2.0

q₀
q₁ ←
q₂
+
c₃

マウスをq1に近づけると、ゴミ箱マークが出てくるので、クリックして消します。

q0だけにして、1量子ビット回路の準備をします。

Probabilities Q-sphere

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[3];
creg c[3];

```

5 $|000\rangle$

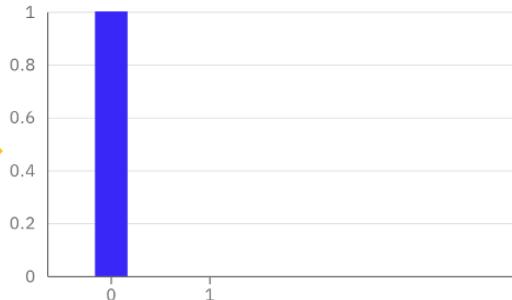


1. Xゲート(NOTゲート)

図の回路を作つてみてください。下に表示される棒グラフの変化を確認しましょう。



初期状態は $|0\rangle$



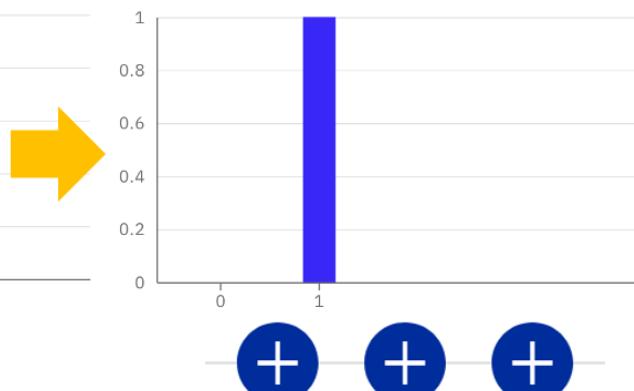
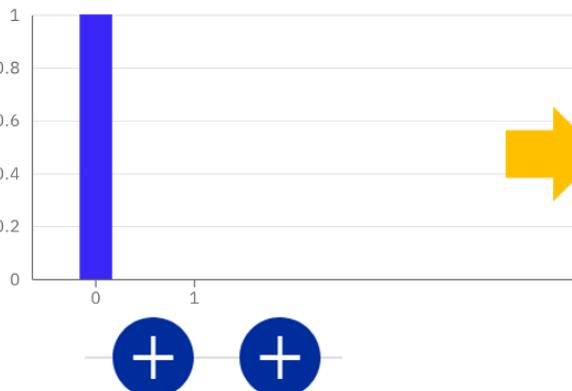
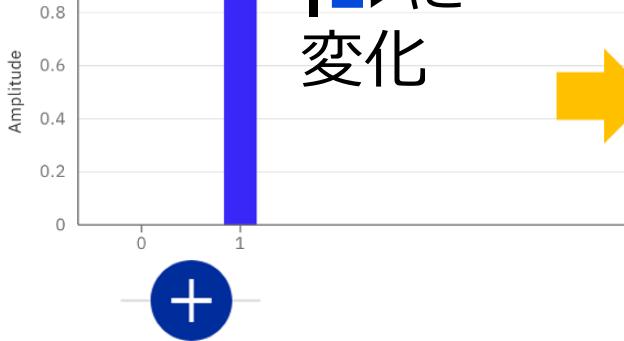
棒グラフ (Statevector 表示) は
量子ビットの状態

$\alpha \times |0\rangle + \beta \times |1\rangle$
の α, β (確率振幅) です。

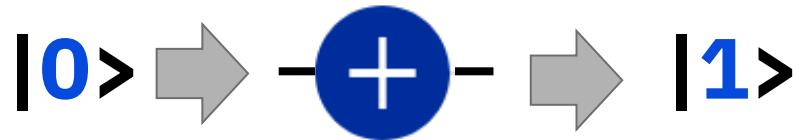
$$|0\rangle = 1 \times |0\rangle + 0 \times |1\rangle$$

$$|1\rangle = 0 \times |0\rangle + 1 \times |1\rangle$$

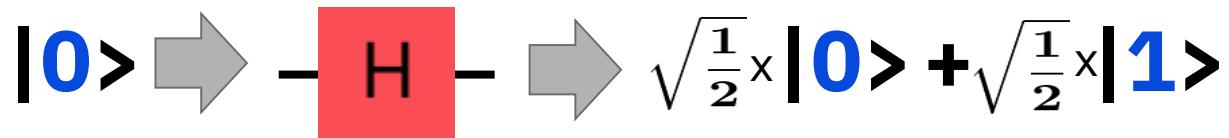
$|1\rangle$ に
変化



量子コンピューターの計算方法

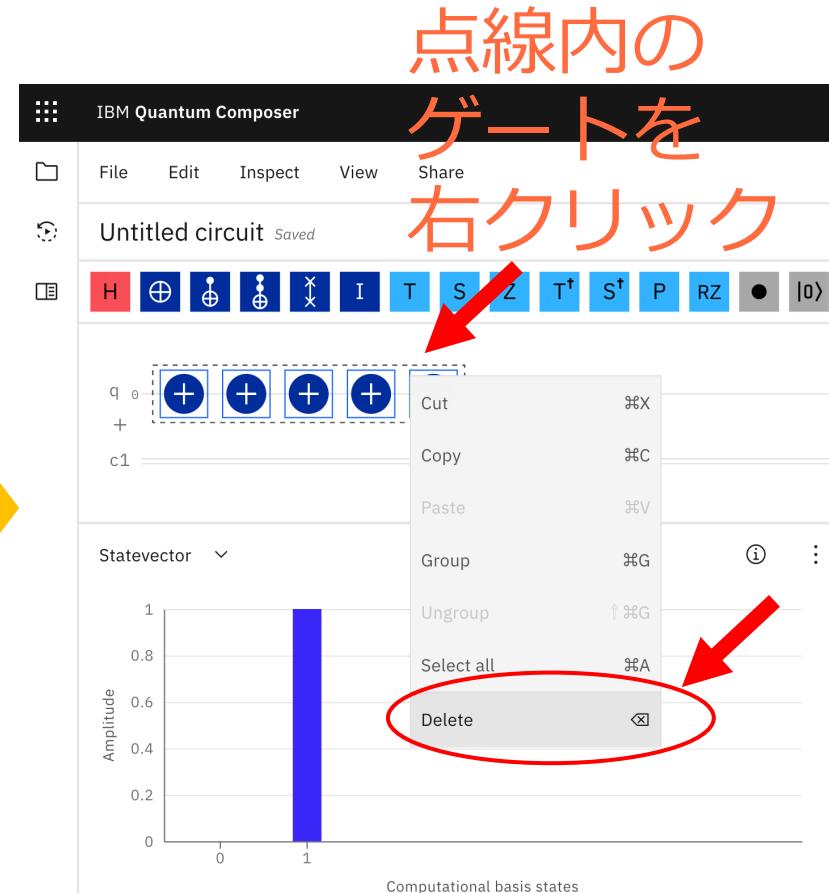
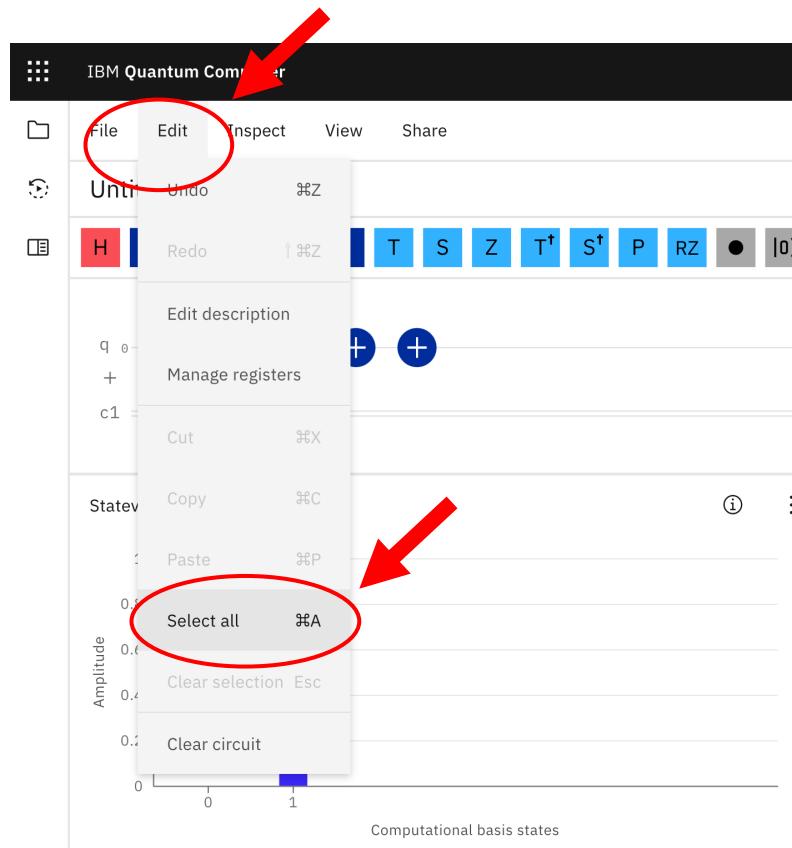


ノット（反転）ゲート



重ね合わせをつくる

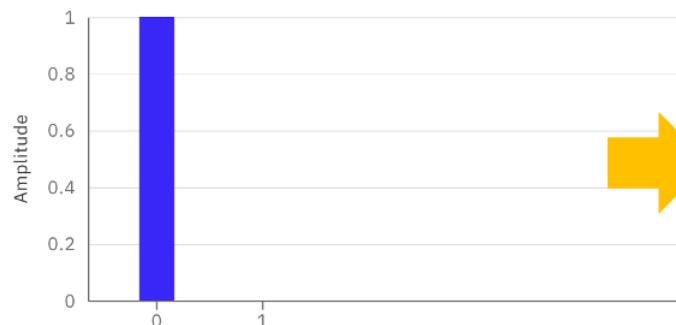
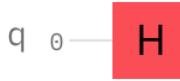
置いたゲートを取り除く



2. Hゲート

図の回路を作ってみてください。下に表示される棒グラフの変化を確認しましょう。

2-1)



重ね合わせ



例えば1000回同じ状態を作って、測定すると約500回は0が観測され、約500回は1が観測される状態。

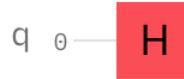
$$|0\rangle = 1 \times |0\rangle + 0 \times |1\rangle$$

$$\begin{aligned} & 0.707 \times |0\rangle + 0.707 \times |1\rangle \\ &= \frac{1}{\sqrt{2}} \times |0\rangle + \frac{1}{\sqrt{2}} \times |1\rangle \end{aligned}$$

2. Hゲート

図の回路を作ってみてください。下に表示される棒グラフの変化を確認しましょう。

2-1)



2-2)



2-3)



量子コンピューターの計算方法

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



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

3. Zゲート

3-1)



3-2)



3-3)



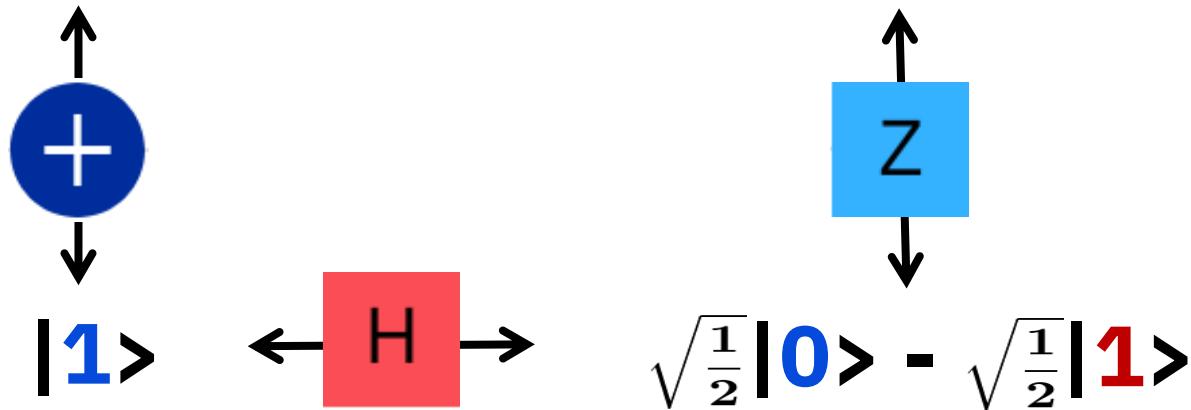
量子コンピューターの計算方法



1 の符号を反転する

量子コンピューターの計算方法

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


$$\begin{array}{c} \uparrow \\ + \\ \downarrow \end{array} \quad \xleftarrow{H} \quad \sqrt{\frac{1}{2}}|0\rangle - \sqrt{\frac{1}{2}}|1\rangle$$

量子ビットを増やす

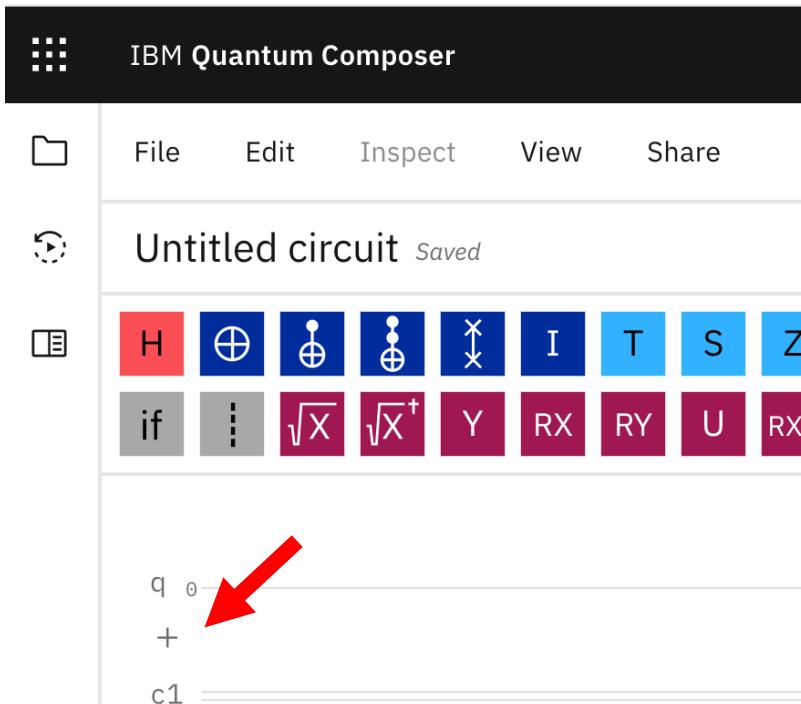
q0の下にある灰色の「+」マークをクリックして2量子ビットの回路を準備します。

IBM Quantum Composer

File Edit Inspect View Share

Untitled circuit Saved

H \oplus \oplus \ominus \ominus \otimes I T S Z
if | \sqrt{X} \sqrt{X}^\dagger Y RX RY U RX



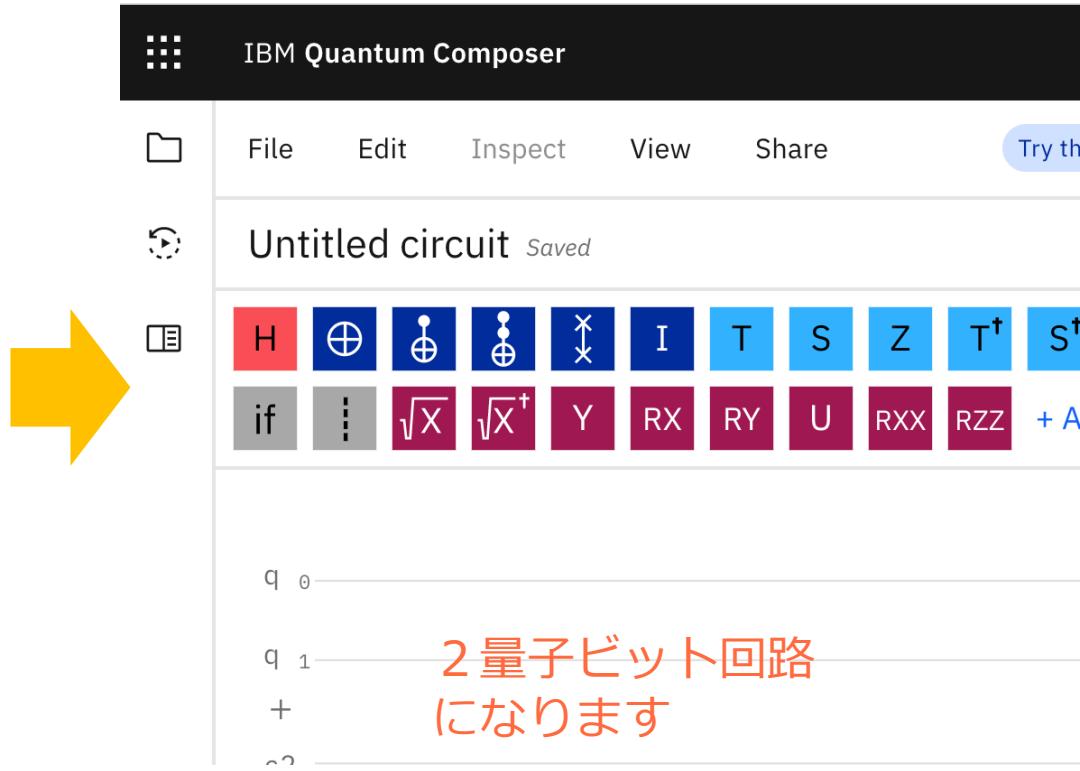
q 0
+
c1 =

IBM Quantum Composer

File Edit Inspect View Share Try th

Untitled circuit Saved

H \oplus \oplus \ominus \ominus \otimes I T S Z T^\dagger S^\dagger
if | \sqrt{X} \sqrt{X}^\dagger Y RX RY U RXX RZZ + A



q 0

q 1
+
c2 =

2量子ビット回路
になります

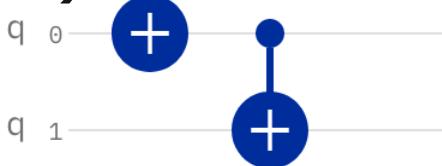
4. CNOTゲート(制御Xゲート)

制御ビットが $|1\rangle$ のときのみ、目標ビットを反転 (NOT) する

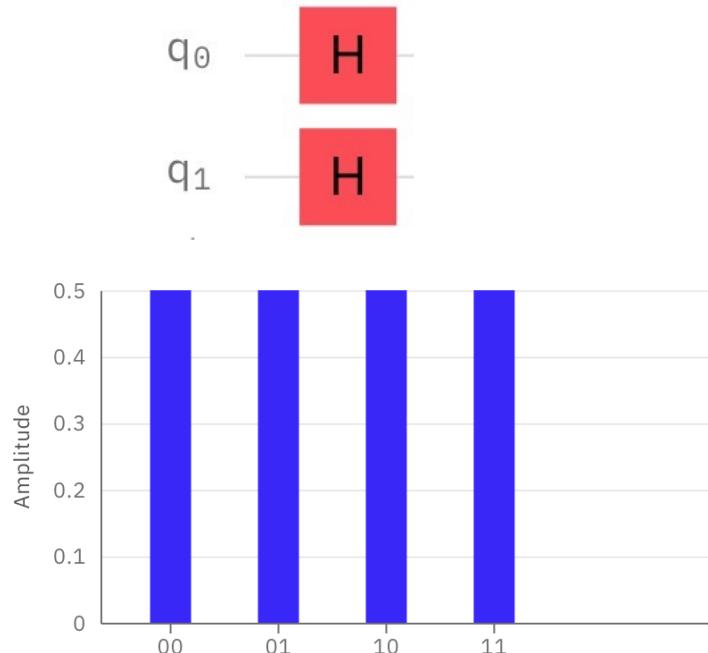


入力		出力	
制御ビット	目標ビット	制御ビット	目標ビット
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

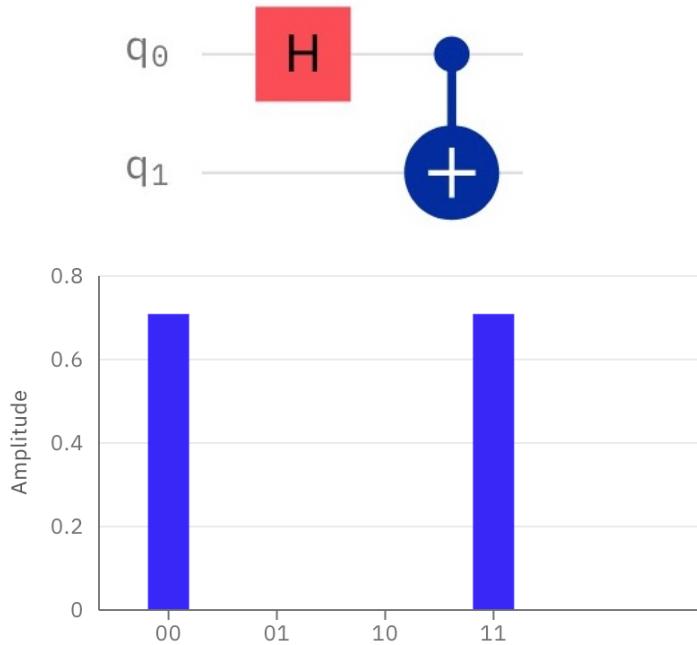
4-1)



量子重ね合わせ

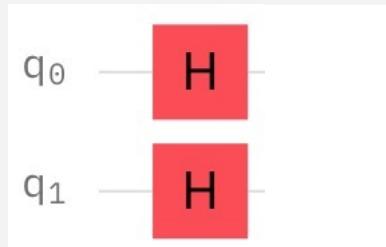


量子もつれ (エンタングルメント)



CNOTゲートは、
エンタングルメントを作ります。

量子重ね合わせ



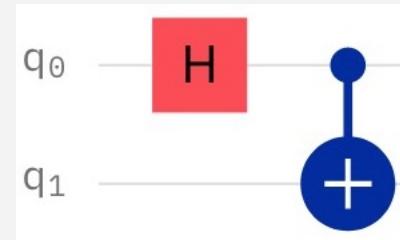
0 0 ... 25%

0 1 ... 25%

1 0 ... 25%

1 1 ... 25%

量子もつれ (エンタングルメント)



0 0 ... 50%

0 1 ... 0%

1 0 ... 0%

1 1 ... 50%

q_0 が0のときは、 q_1 も必ず0。
 q_0 が1のときは、 q_1 も必ず1。

5. グローバーのアルゴリズム

裏返しで置いたトランプをめくって
目的のカードを探すとき、
古典的手法では、
総当たりで一つ一つ探す方法のため、
平均して、トランプの枚数 N の半分は
裏返してみないといけません。



一方、グローバーのアルゴリズムでは、
おおよそ \sqrt{N} ステップで、
目的のカードを探し出すことができます。

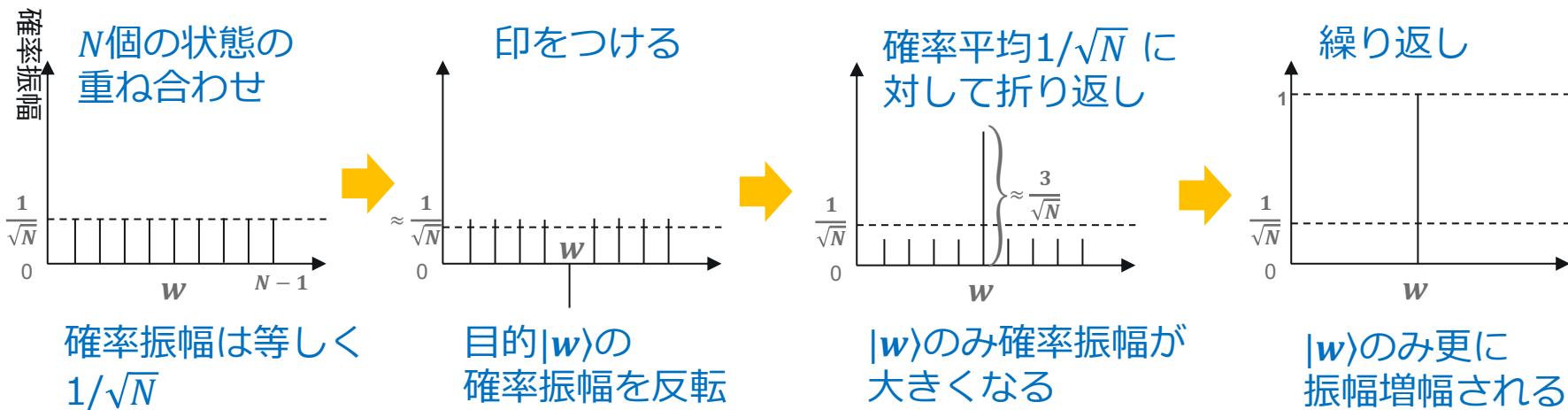
グローバーのアルゴリズム詳細

Step 1: 均一な重ね合わせ状態にする。

Step 2: 目的の状態 $|w\rangle$ に印をつける。(確率振幅を反転。)

Step 3: 平均振幅に対して反転。 $(|w\rangle)$ のみ大きくなる。)

Step 2, 3を繰り返す。

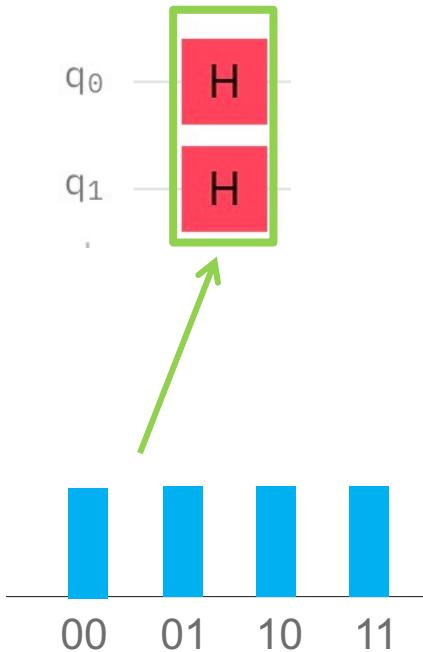


00, 01, 10, 11 から 11 を選ぶアルゴリズム

をまずは作ってみましょう！

00, 01, 10, 11 から 11 を選ぶアルゴリズム

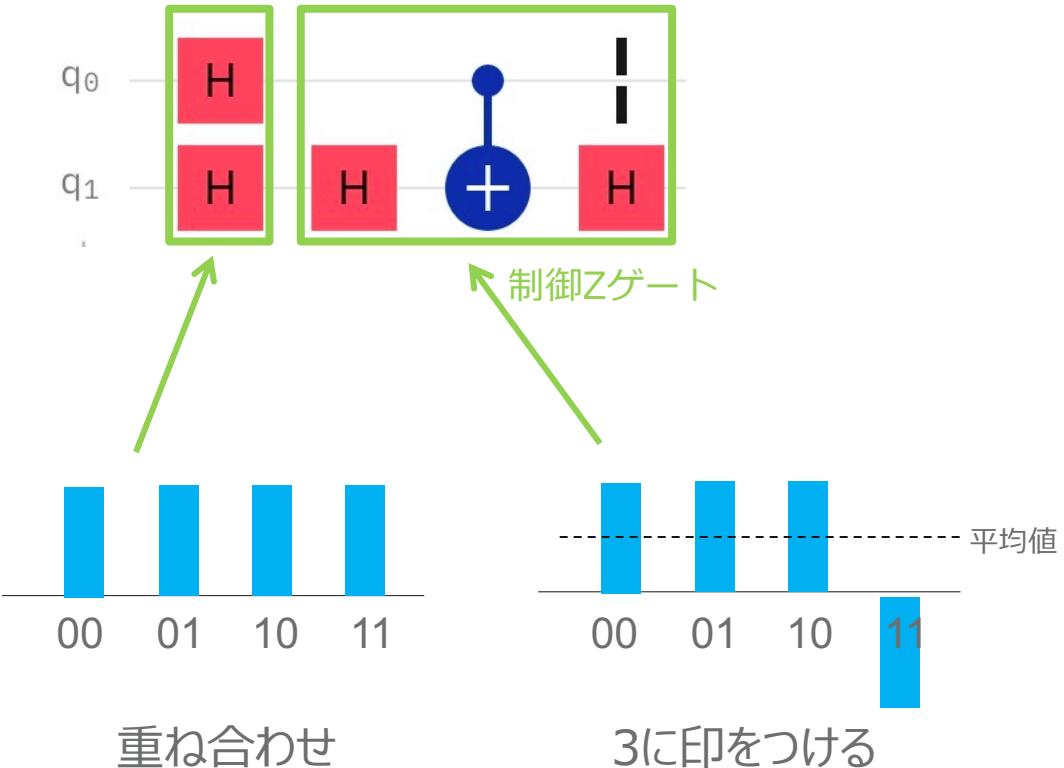
5-1) 重ね合わせ



重ね合わせ

00, 01, 10, 11 から 11 を選ぶアルゴリズム

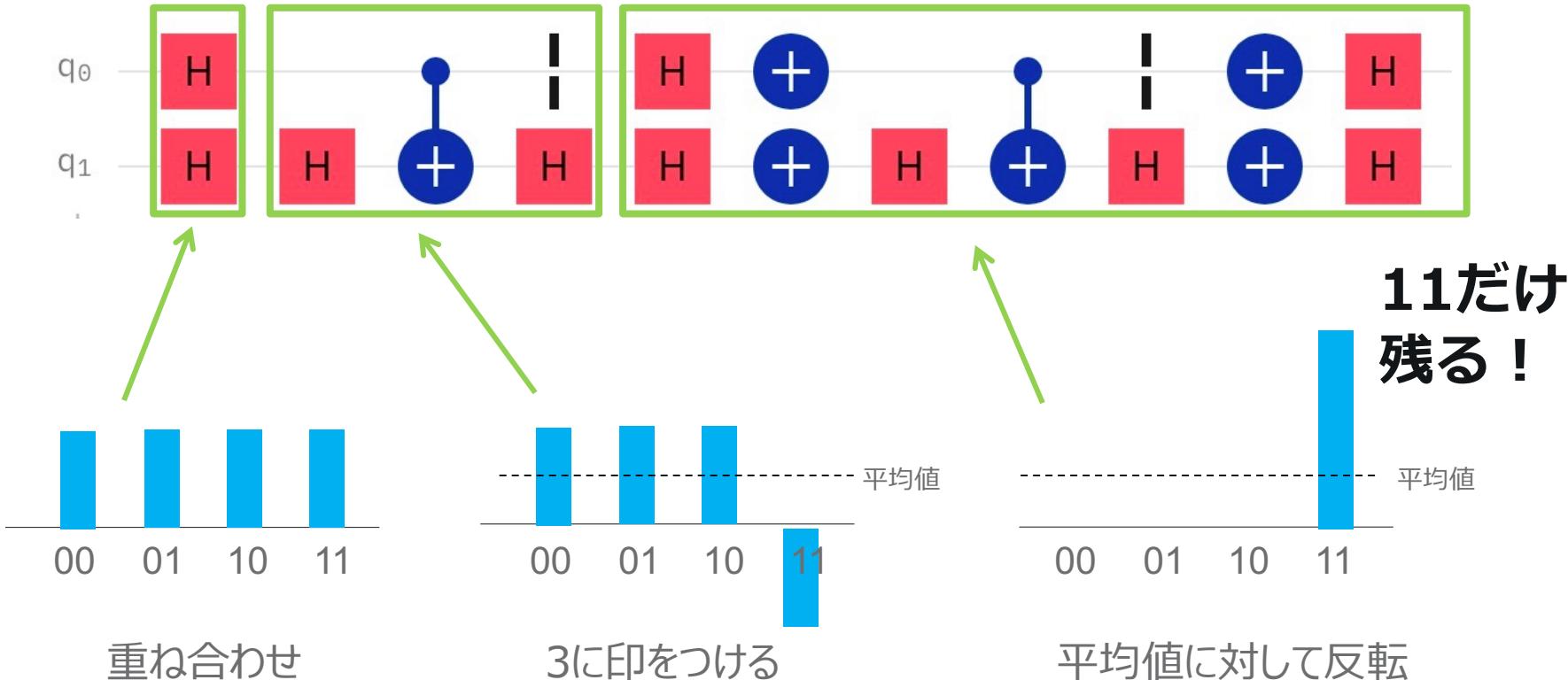
5-2) 11に印をつける



制御Zゲート			
入力		出力	
制御ビット	目標ビット	制御ビット	目標ビット
0	0	0	0
0	1	0	1
1	0	1	0
1	1	1	-1

00, 01, 10, 11 から 11 を選ぶアルゴリズム

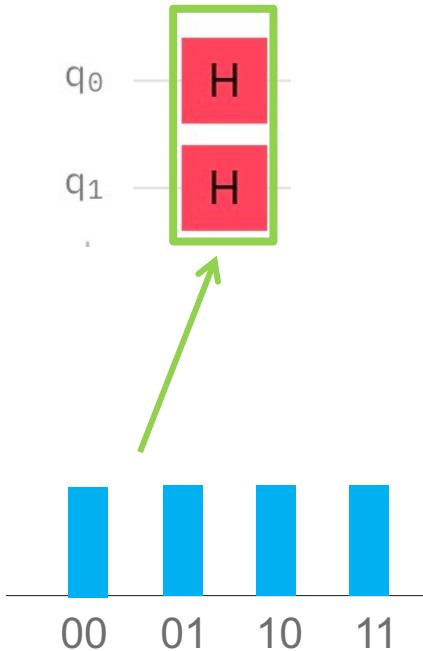
5-3) 平均値に対して反転



00, 01, 10, 11 から 11 を選ぶアルゴリズム

5-1) 重ね合わせ

を見直してみます。

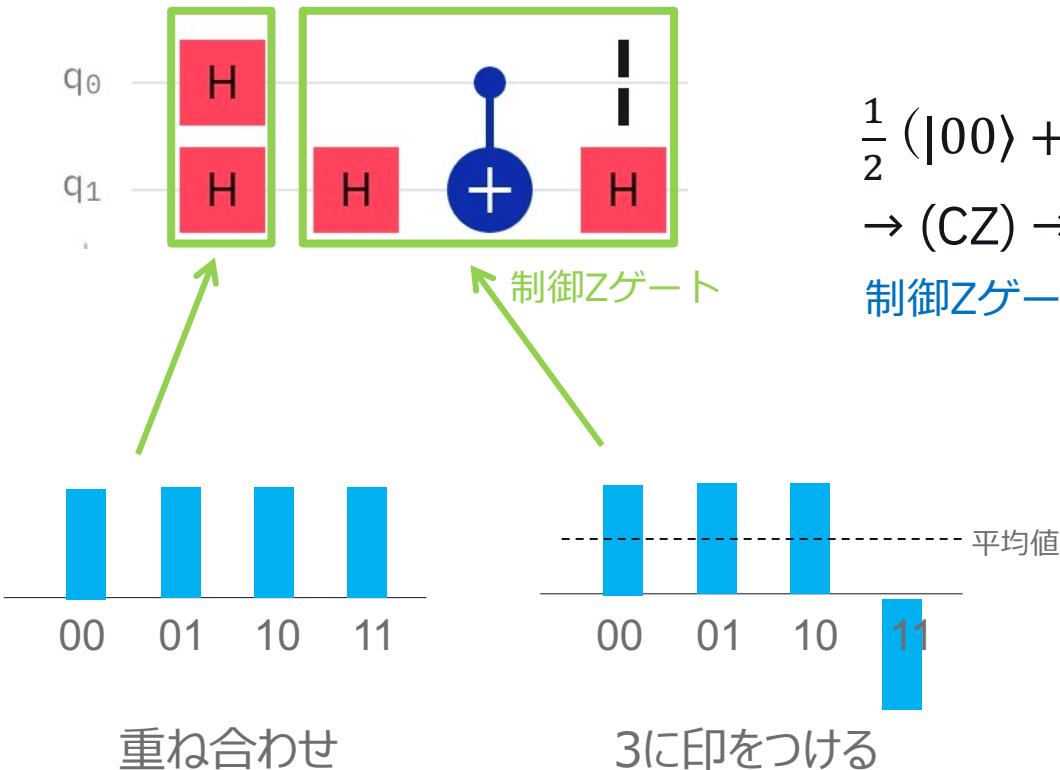


$$\begin{aligned} &|0\rangle|0\rangle \\ &\rightarrow (H \otimes H) \\ &\rightarrow \frac{|0\rangle+|1\rangle}{\sqrt{2}} \frac{|0\rangle+|1\rangle}{\sqrt{2}} \\ &= \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle) \end{aligned}$$

重ね合わせ

00, 01, 10, 11 から 11 を選ぶアルゴリズム

5-2) 11に印をつける



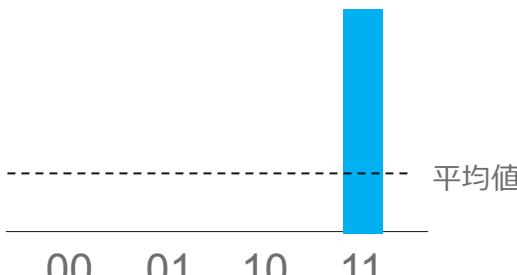
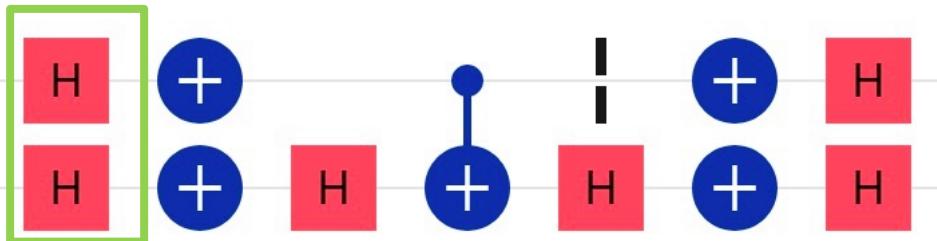
$$\begin{aligned} & \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) \\ \rightarrow & (\text{CZ}) \rightarrow \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle - |11\rangle) \end{aligned}$$

制御Zゲート

|11⟩にのみマイナスが
つきます

00, 01, 10, 11 から 11 を選ぶアルゴリズム

5-3) 平均値に対して反転

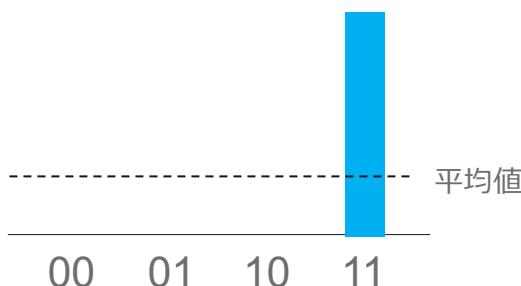
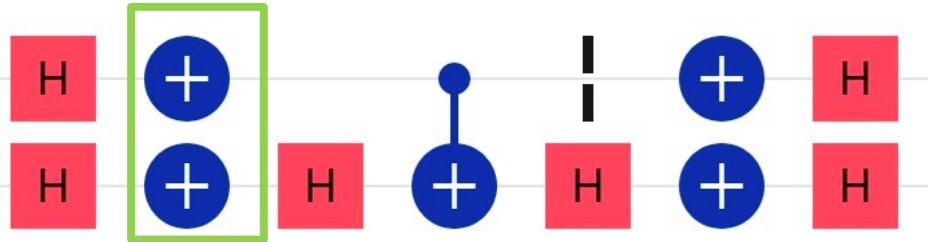


平均値に対して反転

$$\begin{aligned} & \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle - |11\rangle) \\ &= \frac{1}{2} (|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle) \\ &\rightarrow (H \otimes H) \\ &\rightarrow \frac{1}{2} \left\{ \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}} \frac{|0\rangle+|1\rangle}{\sqrt{2}} \right) + \left(\frac{|0\rangle+|1\rangle}{\sqrt{2}} \frac{|0\rangle-|1\rangle}{\sqrt{2}} \right) \right. \\ &\quad \left. + \left(\frac{|0\rangle-|1\rangle}{\sqrt{2}} \frac{|0\rangle+|1\rangle}{\sqrt{2}} \right) - \left(\frac{|0\rangle-|1\rangle}{\sqrt{2}} \frac{|0\rangle-|1\rangle}{\sqrt{2}} \right) \right\} \\ &= (\text{展開省略}) \\ &= \frac{1}{2} (|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle) \end{aligned}$$

00, 01, 10, 11 から 11 を選ぶアルゴリズム

5-3) 平均値に対して反転



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

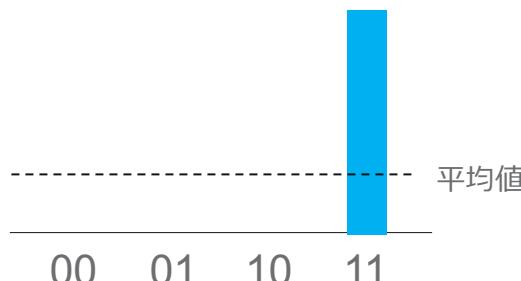
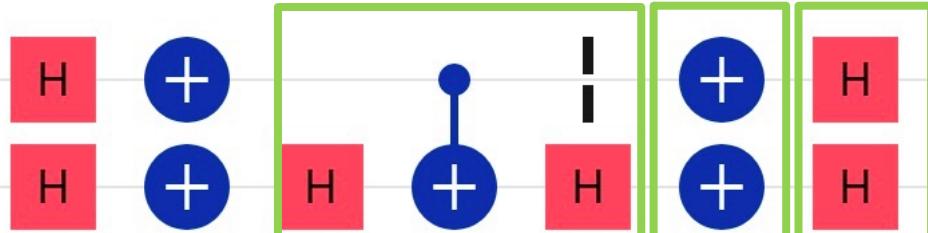
$$\rightarrow (X \otimes X)$$

$$\rightarrow \frac{1}{2} (|1\rangle|1\rangle + |1\rangle|0\rangle + |0\rangle|1\rangle - |0\rangle|0\rangle)$$

平均値に対して反転

00, 01, 10, 11 から 11 を選ぶアルゴリズム

5-3) 平均値に対して反転



平均値に対して反転

$$\begin{aligned} & \frac{1}{2} (|1\rangle|1\rangle + |1\rangle|0\rangle + |0\rangle|1\rangle - |0\rangle|0\rangle) \\ & \rightarrow (CZ) \\ & \rightarrow \frac{1}{2} (-|1\rangle|1\rangle + |1\rangle|0\rangle + |0\rangle|1\rangle - |0\rangle|0\rangle) \\ & \rightarrow (X \otimes X) \\ & \rightarrow \frac{1}{2} (-|0\rangle|0\rangle + |0\rangle|1\rangle + |1\rangle|0\rangle - |1\rangle|1\rangle) \\ & \rightarrow (H \otimes H) \\ & \rightarrow (\text{展開省略}) \\ & = -|1\rangle|1\rangle = -|11\rangle \end{aligned}$$

11だけ
残りました！

6. 測定

IBM Quantum Composer

File Edit Inspect View Share

Grover Saved (2)

Try the new Composer beta → Setup and run (3)

Visualizations 3623

OpenQASM 2.0

Open in Quantum Lab

1 OPENQASM 2.0;
2 include "qelib1.inc";
3
4 qreg q[2];
5 creg c[2];
6
7 h q[0];
8 h q[1];
9 h q[1];
10 cx q[0] q[1].

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[2];
creg c[2];
h q[0];
h q[1];
h q[1];
cx q[0] q[1].
```

まず量子シミュレーターで実験

Set up and run your circuit

Step 1
Choose a system or simulator

Search by system or simulator name ↑ ▽

simulator_extended_stabilizer See details

Simulator status Online
Total pending jobs 0
63 Qubits

ibmq_qasm_simulator See details

Simulator status Online
Total pending jobs 0
32 Qubits

simulator_statevector See details

Simulator status Online
Total pending jobs 0
32 Qubits

Close

Run on ibmq_qasm_simulator

(4) スクロール

(5)

(6)

ibmq_qasm_simulator
を選択



Jobs /
6239316319e689953fc800f6

See more details ↗

Completed
Mar 22, 2022 11:16 AM (in 9.5s)

Backend
ibmq_qasm_simulator

Status timeline Completed

Details

Result - histogram



11だけが
観測されました！

File Edit Inspect View Share

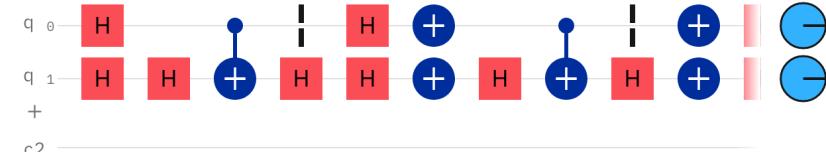
Try the new Composer interface →

Setup and run ↗

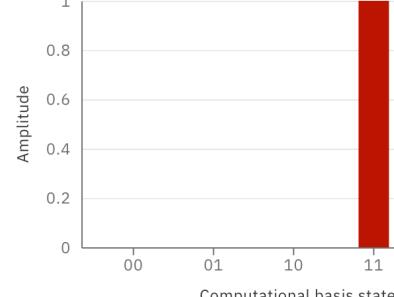
Grover Saved



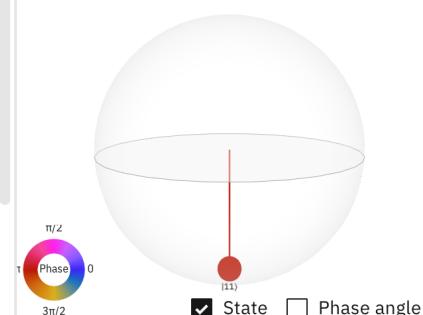
+ Add



Statevector



Q-sphere



OpenQASM 2.0 ▾

Open in Quantum Lab

```

1 OPENQASM 2.0;
2 include "qelib1.inc";
3
4 qreg q[2];
5 creg c[2];
6
7 h q[0];
8 h q[1];
9 h q[1];
10 cx q[0],q[1];
11 barrier q[0];
12 h q[1];
13 h q[0];
14 h q[1];
15 x q[0];
16 x q[1];
17 h q[1];
18 cx q[0],q[1];
19 barrier q[0];
20 h q[1];
21 x q[0];
22 x q[1];
23 h q[0];
24 h q[1];
25 measure q[0] -> c[0];
26 measure q[1] -> c[1];

```

(10)

Visualizations seed

3623

Set up and run your circuit

実機の量子コンピューターで実験

X

Step 1

Choose a system or simulator

Search by system or simulator name

ibmq_lima

System status ● Online

Total pending jobs 23

5 Qubits 8 QV 2.7K CLOPS

(11)



Step 2

Choose your settings

Provider

ibm-q/open/main

実デバイスを選択します。

- Pending jobsが少ない
- QVが大きい

などから選んでみましょう。

ibmq_armonk

System status ● Online

Total pending jobs 0

1 Qubit 1 QV

simulator_stabilizer

Simulator status ● Online

Total pending jobs 5

5000 Qubits

e.g. Untitled circuit job

Tags

Add tags

Close

Run on ibmq_lima

(12)

Jobs /
6239316319e689953fc800f6

(13) Jobの確認

See more details ↗

Completed
Mar 22, 2022 11:16 AM (in 9.5s)

Backend
ibmq_qasm_simulator

Status timeline

Completed



Grover Saved



+ Add



Try the new Composer beta →

Setup and run ↗

Visualizations seed

3623

OpenQASM 2.0 ←

Open in Quantum Lab

```
1 OPENQASM 2.0;
2 include "qelib1.inc";
3
4 qreg q[2];
5 creg c[2];
```

IBM Quantum Composer

View all jobs →

File Edit Inspect View Share

Try the new Composer beta →

Setup and run ↗

Composer jobs

Search jobs

current-file X

(14)

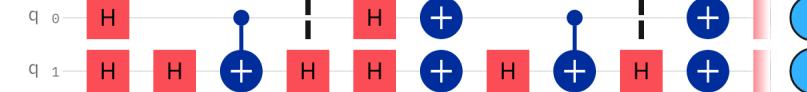
Pending: Mar 22, 2022 11:18 AM
ID: 623931f88293e91a6f1e5075 | ibmq_lima

Completed: Mar 22, 2022 11:16 AM
ID: 6239316319e6899... | ibmq_qasm_simulator

Grover Saved



+ Add



Visualizations seed

3623

OpenQASM 2.0 ←

Open in Quantum Lab

```
1 OPENQASM 2.0;
2 include "qelib1.inc";
3
4 qreg q[2];
5 creg c[2];
6
7 h q[0];
8 l [1];
```

IBM Quantum Composer

Jobs / 623931f88293e91a6f1e5075 See more details

Estimated time to completion in 28 minutes

Backend ibmq_lima

Status timeline

- Created: Mar 22, 2022 11:18 AM
- Transpiling: 885ms
- Validating: 1.2s
- In queue
- Running expected to run in 28 minutes, queue pos: 5
- Completed

File Edit Inspect View Share Try the new Composer beta → Setup and run

Saved

+ Add

OpenQASM 2.0

Open in Quantum Lab

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[2];
creg c[2];
h q[0];
h q[1];
h q[1];
cx q[0],q[1];
barrier q[0];
h q[1];
h q[0];
h q[1];
v q[0].
```

Diagram illustrating a quantum circuit:

- Two qubits (q₀ and q₁) and one classical register (c2) are involved.
- The circuit starts with Hadamard (H) gates on both qubits.
- Following the initial H gates, there are CNOT gates between q₀ and q₁, and between q₁ and q₀.
- After the CNOT gates, another Hadamard (H) gate is applied to each qubit.
- Finally, a CNOT gate is applied between q₀ and q₁.

IBM Quantum Composer

Jobs / 622ed5b36858d0f1dfc13c20

Completed Mar 14, 2022 2:42 PM (in 13s)

Backend ibmq_santiago

Status timeline

Details

Result - histogram

Measurement outcome

Frequency

(15)

File Edit Inspect View Share Try the new Composer beta → Setup and run

Grover Saved Visualizations seed 3623

+ Add

Quantum circuit diagram:

```
graph LR; q0[ ] -- H --> q0[ ]; q0[ ] --+--> q1[ ]; q1[ ] -- H --> q1[ ]; q1[ ] --+--> q0[ ]; q0[ ] -- H --> q0[ ]; q0[ ] --+--> q1[ ]; q1[ ] -- H --> q1[ ]; q1[ ] --+--> q0[ ]; q0[ ] -- H --> q0[ ]; q0[ ] --+--> q1[ ]; q1[ ] -- H --> q1[ ]; q1[ ] --+--> q0[ ]; q0[ ] -- H --> q0[ ]; q0[ ] --+--> q1[ ]; q1[ ] -- H --> q1[ ]; q1[ ] --+--> q0[ ];
```

Statevector

Q-sphere

Amplitude

Computational basis states

Phase angle

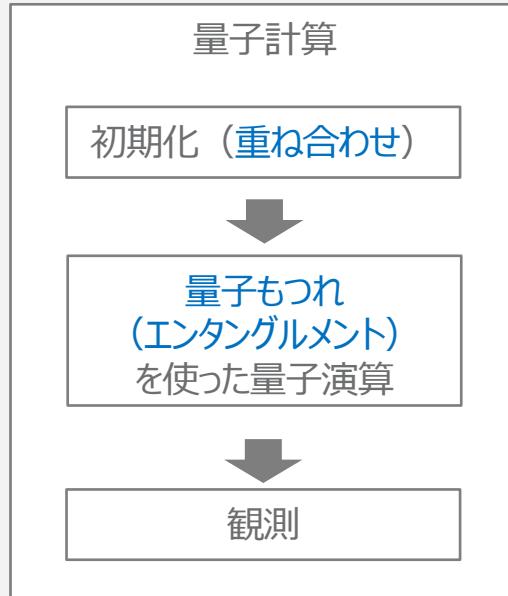
OpenQASM 2.0

```
OPENQASM 2.0;
include "qelib1.inc";
qreg q[2];
creg c[2];
h q[0];
h q[1];
h q[1];
cx q[0],q[1];
barrier q[0];
h q[1];
h q[0];
h q[1];
x q[0];
x q[1];
h q[1];
h q[1];
cx q[0],q[1];
barrier q[0];
h q[1];
x q[0];
x q[1];
h q[0];
h q[1];
measure q[0] -> c[0];
measure q[1] -> c[1];
```

Visualizations seed 3623

Open in Quantum Lab

量子アルゴリズム



重ね合わせの状態では全ての出現確率は同じ



量子ビットを干渉させる演算



特定のパターンの
確率のみを高める



量子状態の重ね合わせ（並列計算）を
干渉（量子もつれ・エンタングルメント）させて
ほしい解を取り出す。

応用例 IBM Quantum Challenge 2019

自治体のコンビニ出店プランを提示せよ

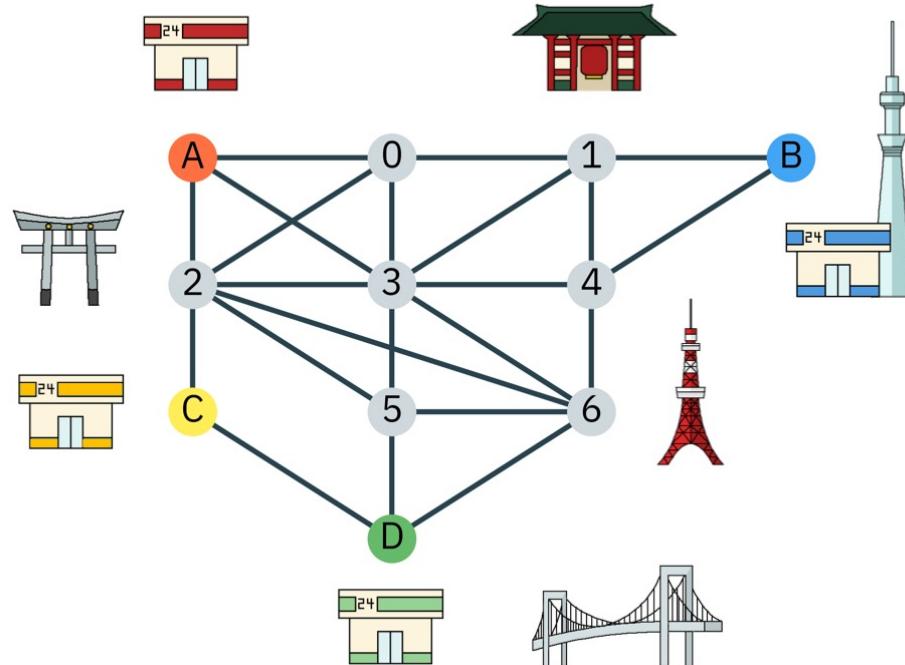
東京のZ市は11の区域からなる自治体で、すでに4社のコンビニ(A、B、C、D社)が本社の1店舗を別々の区域に展開しています。

市長のあなたは、残りの7区域にもコンビニを誘致しようとしたが、4社から以下の条件が提示されました。

- ・1つの区域に出店出来るのは1社のコンビニのみ
- ・自社のコンビニは、隣接する区域に自社のコンビニが既に出店している場合は出店しない。

あなたはこれらの条件を満たす出店案を提示できるでしょうか？

グローバーのアルゴリズムを用いて、条件を満たす全ての出店案を列挙してみてください。



<https://github.com/quantum-challenge/2019/blob/master/problems/final/Final.ipynb>

続きは、Qiskitテキストブックで！



Qiskitを使った量子計算の学習

量子とは？

0. 前提条件

1. 量子状態と量子ビット

1.1 はじめに

1.2 計算の原子

1.3 量子ビット状態を表現する

1.4 単一量子ビットゲート

1.5 量子コンピューターの場合

2. 復数量子ビットともつれ状態

2.1 はじめに

2.2 復数量子ビットともつれ状態

2.3 位相キックバック

2.4 さらなる回路の等価性

2.5 普遍性の証明

2.6 量子コンピューター上の古典計算

3. 量子プロトコルと量子アルゴリズム

3.1 量子回路

3.2 ドイチ-ヨサのアルゴリズム

3.3 ペルンシュタイン・ヴァジラニアル

Japanese

The new Qiskit Textbook beta is now available. Try it out now

Qiskit を使った量子計算の学習

Qiskit Communityチームからのご挨拶です！Qiskitをベースとした大学の量子アルゴリズム/計算コースの補足教材となるよう、このテキストブックを作り始めました：

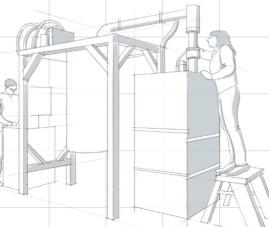
- 量子アルゴリズムの基礎となる数学
- 今日の非フォールトトレントな量子デバイスの詳細
- IBMのクラウド型量子システムに量子アルゴリズムを実装するためのQiskitでのコーディング

テキストブックを読む →

Overview Learn Community Documentation

Qiskit

Overview Learn



→ 5. グローバーのアルゴリズムで数独を解く

この章でこれまで使われていたオラクルは、事前にその解が分かっているものから作成されています。ここでは、グローバーのアルゴリズムを使用して、事前に解を知らないでも解ける単純な問題を解きます。その問題は 2×2 のバイナリーの数独で、以下の2つのシンプルなルールに基づいています：

- 同じ値を2回含む列はない
- 同じ値を2回含む行はない

数独の各正方形を次の図のような変数に割り当てて：

V_0	V_1
V_2	V_3

<https://qiskit.org/textbook/ja/preface.html>

回路にこの数独の解を出力させたいと思います。

Qiskit：より高度なアルゴリズムの実装に

The screenshot shows the IBM Quantum Composer interface. On the left, a quantum circuit diagram for a Grover search algorithm is displayed on three qubits (q_0, q_1, q_2) and one classical register (c2). The circuit consists of several Hadamard (H), controlled-phase (+), and CNOT gates. In the center, the circuit's OpenQASM 2.0 source code is shown, with a red box highlighting the Qiskit imports and the first few lines of the circuit definition. On the right, the Qiskit visualization pane is open, showing the Qiskit circuit object with a red box around the "Open in Quantum Lab" button. A large red arrow points from the "Open in Quantum Lab" button to the text "Open in Quantum Lab" on the right. Another red arrow points from the Qiskit imports in the OpenQASM code to the Qiskit imports in the visualization pane.

```
OpenQASM 2.0
OpenQASM 2.0
Qiskit
    );
    .lib1 inc";
4  qreg q[2];
5  creg c[2];
6
7  h q[0];
8  h q[1];
9  h q[1];
10 cx q[0],q[1];
11 barrier q[0];
```

Open in Quantum Lab

IBM Quantum Lab

量子計算用開発キットQiskitを インストールなしで実行できる Jupyter Notebook環境

本日の内容

- 量子コンピューター入門
- IBM Quantum Composer ハンズオン
 - 基本の量子ゲートの操作
 - グローバーのアルゴリズムの実装
 - 実機量子コンピューターでの実験
- 今日の資料：
https://github.com/quantum-tokyo/qiskit-handson/composer/20220331_IQX_grover.pdf

Qiskitコミュニティによる おすすめイベント・教材

最新のご案内はTwitterから  @qiskit

日本語リンク集：<https://github.com/quantum-tokyo/introduction>

特別デバイス

- 研究者プログラム
- 教育用プログラム

オンライン教材・翻訳活動

- Qiskit テキストブック
- Qiskit ドキュメント チュートリアル
- 翻訳プロジェクト

イベント

- 量子プログラミングコンテスト
- サマースクール
- オープン・サイエンス・アワード

認定制度

- Qiskit Advocate プログラム
- Qiskit デベロッパー認定制度

Qiskitコミュニティメンバーの協力によって運営・開催されています！
みなさんもぜひご参加ください！

ありがとうございました。

ワークショップ、セッション、および資料は、IBMまたはセッション発表者によって準備され、それぞれ独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる参加者に対しても法律的またはその他の指導や助言を意図したものではなく、またそのような結果を生むものではありません。本講演資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本講演資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本講演資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引きだすことを意図したものでも、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでもなく、またそのような結果を生むものではありません。

本講演資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本講演資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもつていっても変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本講演資料に含まれている内容は、参加者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したものでも、またそのような結果を生むものではありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプロセッシングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、IBM Q、Qiskit は、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtmlをご覧ください。