Qiskitテキストブック勉強会
「量子ウォークによる探索アルゴリズム」

門脇 正史
DENSO & AIST

量子ウォークによる探索アルゴリズム

https://ja.learn.qiskit.org/course/ch-algorithms/quantum-walk-search-algorithm

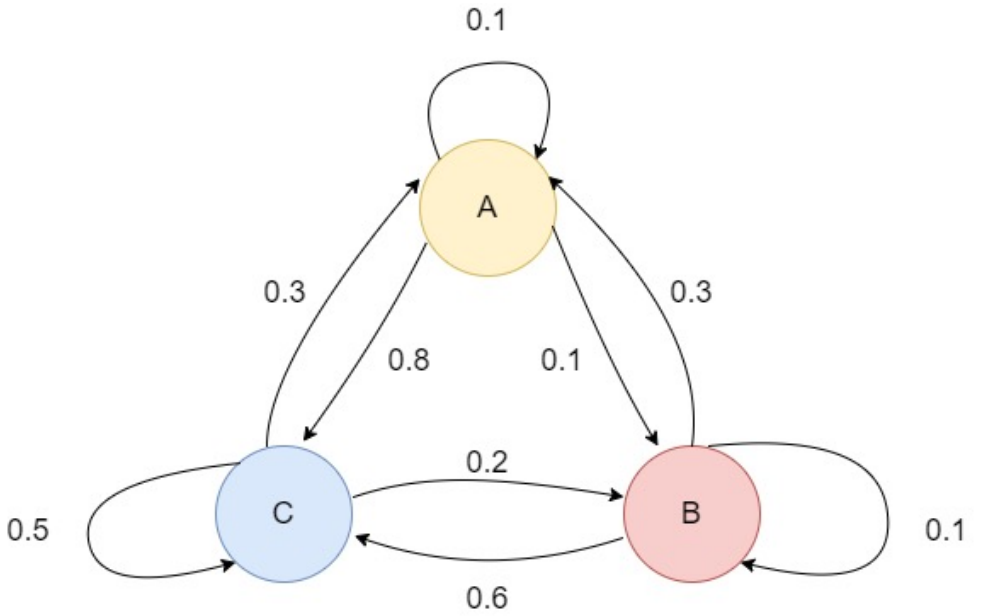
1. 古典マルコフ連鎖
2. 量子ウォーク
   - 2.A コイン量子ウォーク
   - 2.B セゲディ量子ウォーク
   - 2.C コインとセゲディ量子ウォークの等価性
3. 超立方体上の量子ウォーク
4. 量子ウォークによる探索アルゴリズム
5. 4次元超立方体上の量子ウォークによる探索
6. 参考文献

1. 古典マルコフ連鎖

マルコフ連鎖はマルコフ性を満たす。つまり次のステップでの確率過程は現在のステップのみに依存して、それ以前には依存しない。ランダムウォークもマルコフ連鎖の一つ。

A,B,Cの状態間の遷移確率の例

遷移行列　$P = \begin{pmatrix} 0.1 & 0.3 & 0.3 \\ 0.1 & 0.1 & 0.2 \\ 0.8 & 0.6 & 0.5 \end{pmatrix}.$



2. 量子ウォーク

量子ウォークは古典のマルコフ連鎖の量子版。

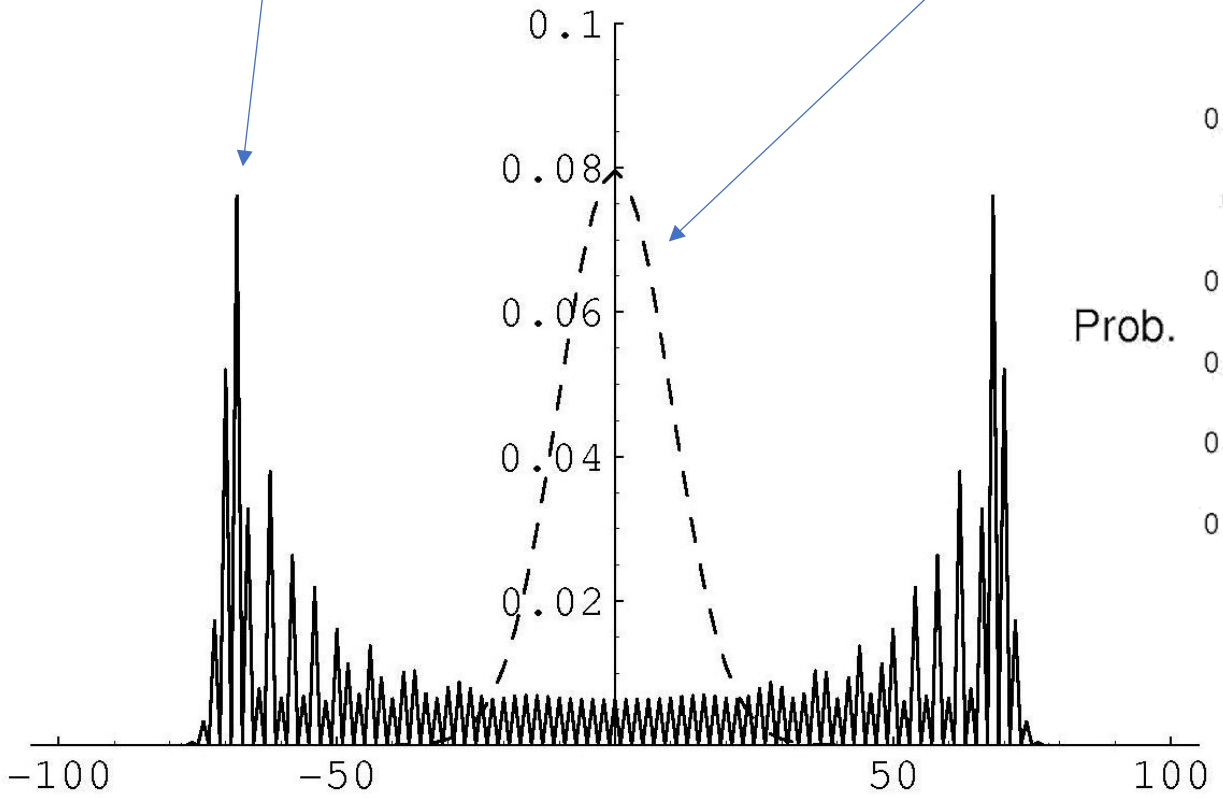
遷移行列ではなくユニタリ行列を定義する。

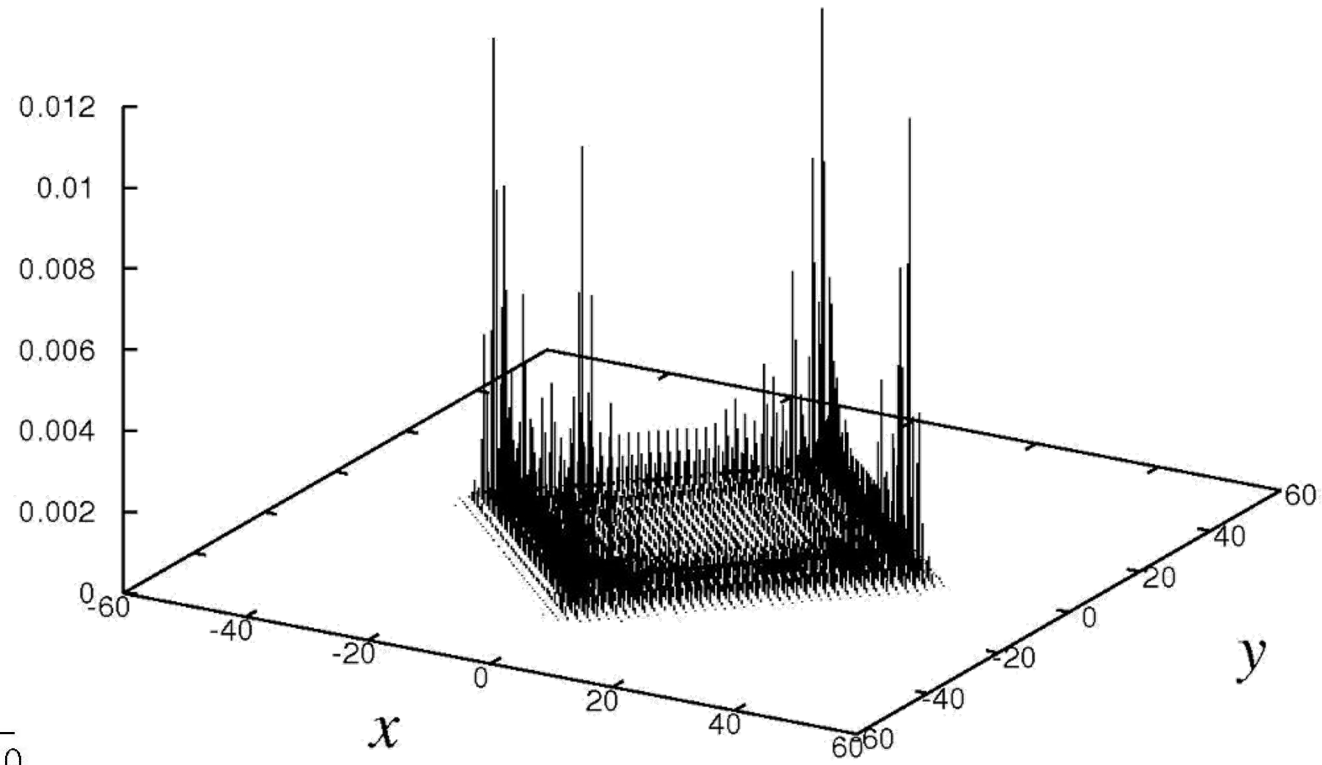
$$P_{ij} \to \sqrt{P_{ij}}$$

以後、離散時間コイン量子ウォーク、セゲディ量子ウォークを順に解説。

(解説は、一次元無限長の整数上や超立方体上で解説するが、より複雑なグラフ上でも考えることができる。)

量子ウォーク　　ランダムウォーク





https://rad-it21.com/サイエンス/konno_norio_20200422/

2.A コイン量子ウォーク

無限長の整数上の運動を考える。

整数$\mathbb{Z}$上での位置を$\{|j\rangle : j \in \mathbb{Z}\}$とし、コインの計算基底は$\{|0\rangle, |1\rangle\}$とする。

コインが$|0\rangle$の場合はある方向に動き、コインが$|1\rangle$の場合はもう一方に動く。

$$\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_P$$

(コインの状態空間 $\otimes$ 位置の状態空間)

- コインの状態は、コイン演算子Cにより作る。
- 位置の状態は、(コインの状態に依存する)シフト演算子Sにより作る。

アダマールコイン

グローバーコイン

$$H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} = 2|s\rangle\langle s| - I, \qquad |s\rangle = \frac{1}{\sqrt{N}}\sum_{i}^{N}|i\rangle$$

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

$$G = \begin{bmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \cdots & \frac{2}{N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{N} & \frac{2}{N} & \cdots & \frac{2}{N} - 1 \end{bmatrix}.$$

等確率

$|000\rangle$により大きな確率を持つ

シフト演算子は$\mathcal{H}_P$に作用して、次の位置に移動させる。整数上の運動では、
シフト演算子により、コインが$|0\rangle$の時は右に動き、コインが$|1\rangle$の時は左に動く:

$$S|0\rangle|j\rangle = |0\rangle|j+1\rangle$$
$$S|1\rangle|j\rangle = |1\rangle|j-1\rangle$$

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

一ステップ分のコイン量子ウォークを以下のユニタリ演算子Uとして表現できる

$$U = SC$$

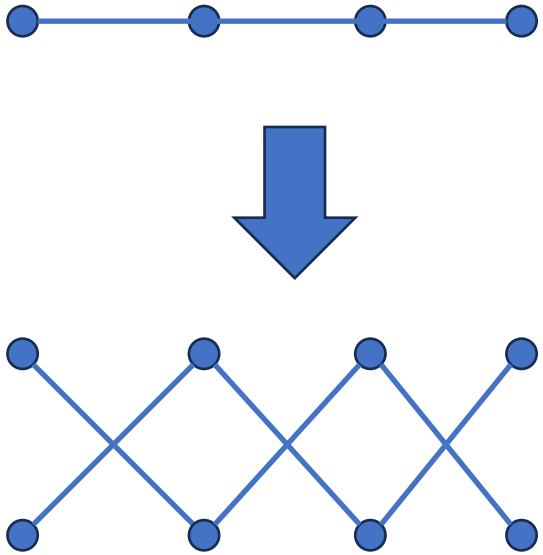
t時間ステップ後の量子状態$|\psi(t)\rangle$は以下で表すことができる

$$|\psi(t)\rangle = U^t|\psi(0)\rangle$$

ここで、$|\psi(0)\rangle$は初期状態。

2.B セゲディ量子ウォーク

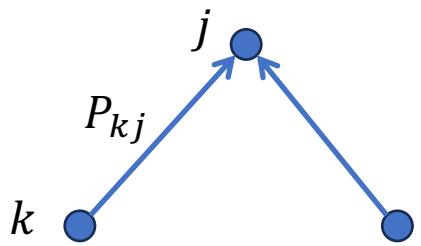
セゲディ量子ウォークは元のグラフの二部二重被覆グラフ上の辺を動く。



任意の$N\times N$の遷移行列Pを持つN-頂点グラフに対して、離散時間量子ウォークをヒルベルト空間$\mathcal{H}^N \otimes \mathcal{H}^N$上のユニタリ演算子として定義する。正規化された状態は以下で定義できる。

$$|\psi_j\rangle \coloneqq \sum_{k=1}^{N} \sqrt{P_{kj}}|j,k\rangle, \qquad j = 1,\dots,N$$

ここでP_{jk}はjからkへの遷移確率。



$\mathcal{H}^N$: N個の基底をもつヒルベルト空間(?)

$|\psi_j\rangle$への射影は

$$\Pi \coloneqq \sum_{j=1}^{N} |\psi_j\rangle\langle\psi_j|$$

シフト演算子は

$$S \coloneqq \sum_{j,k}^{N} |j,k\rangle\langle k,j|$$

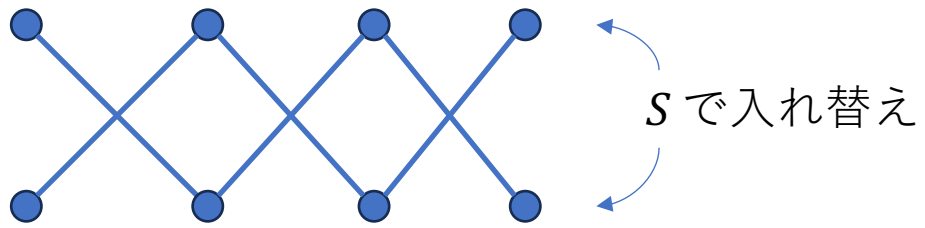
量子ウォークは

$$U \coloneqq S(2\Pi - I)$$

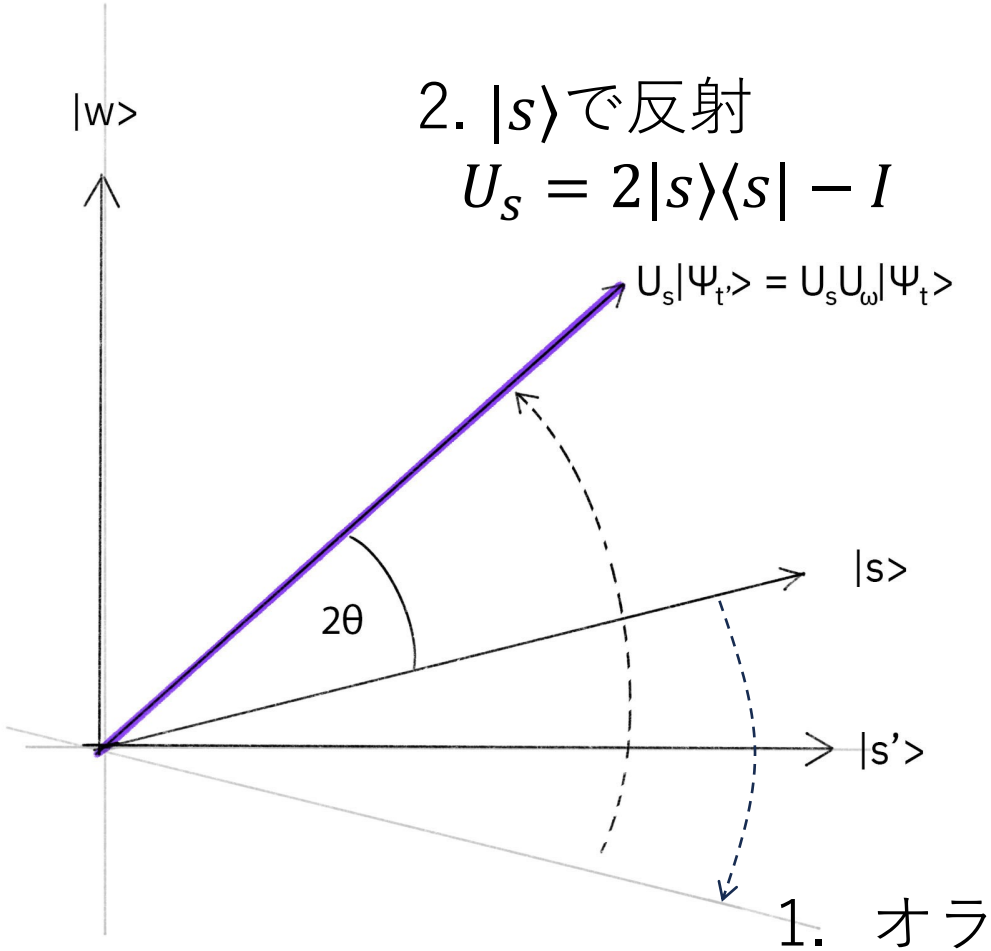
したがって、t時間ステップ後の量子状態$|\psi(t)\rangle$は

$$|\psi(t)\rangle = U^t|\psi(0)\rangle$$

$2\Pi - I$：反射演算子
グラフが繋がっている
エッジをマーク
（コインや グローバー拡散演算子と同じ）

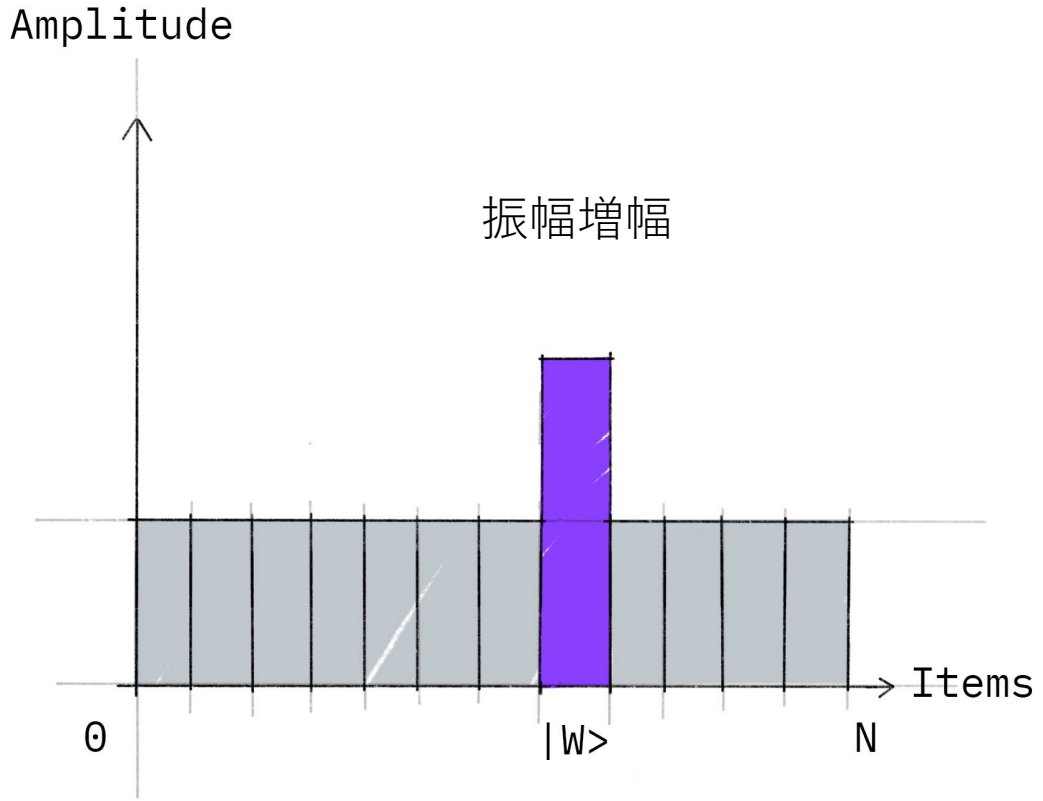


ここで少し、グローバー探索のおさらい



2. $|s\rangle$で反射

$$U_s = 2|s\rangle\langle s| - I$$



1. オラクルによるマーク
（$|s'\rangle$で反射）

https://ja.learn.qiskit.org/course/ch-algorithms/grovers-algorithm

2.C コインとセゲディ量子ウォークの等価性

Quantum Inf Process (2017) 16:215
DOI 10.1007/s11128-017-1667-y

CrossMark

Equivalence of Szegedy's and coined quantum walks

Thomas G. Wong[1,2]

イントロより

Under certain conditions, it is known from Fact 3.2 of[18] that one step of Szegedy's quantum walk is equivalent to two steps of the coined quantum walk. This fact is stated without proof, however, and the precise relationships between the individual operators are not given. Some details are described in lecture notes [19], but again the exact relationships between individual operators are not explored. In this paper, we amend this oversight, explicitly showing that Szegedy's first reflection operator is equal to the "Grover diffusion" coin flip of the coined quantum walk, while Szegedy's second reflection operator is equal to the combination of a "flip-flop" shift, Grover diffusion coin flip, and another flip-flop shift. These relationships between the operators also hold for the seminal search schemes, where Szegedy's quantum walk quantizes a Markov chain with absorbing marked vertices [14], and the coined quantum walk uses the Grover diffusion coin for unmarked vertices and the negative identity for marked vertices [1].

3. 超立方体上の量子ウォーク

(チュートリアルでは4次元だが、このページのみ3次元で説明)

3次元立方体の場合、000は001, 010, 100と隣接する。異なるビットの位置を a とラベルする(コイン)。ヒルベルト空間は

$$\mathcal{H} = \mathcal{H}^n \otimes \mathcal{H}^{2^n}, \qquad n = 3$$

(コインの状態空間 $\otimes$ 位置の状態空間)

計算基底は

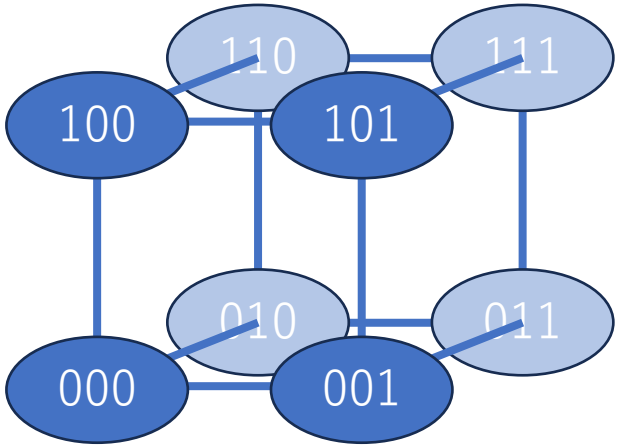
$$\{|a\rangle|v\rangle \mid a \in \{0,1,2\}, v \in \{000, \ldots, 111\}\}$$

シフト演算子は

$$S|a\rangle|v\rangle = |a\rangle|v \oplus e_a\rangle$$

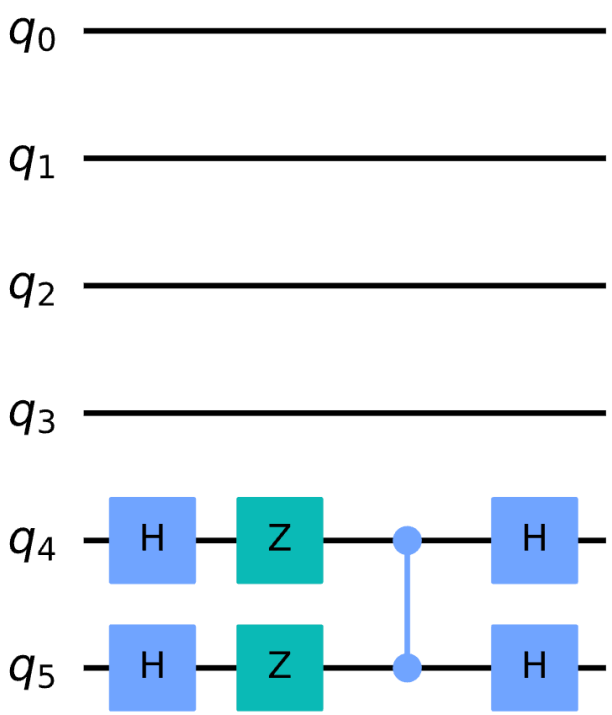
グローバーコインGを用いて、発展演算子は

$$U = SG$$



グローバーコイン(グローバー拡散演算子) G

```
one_step_circuit = QuantumCircuit(6, name=' ONE STEP')
# Coin operator
one_step_circuit.h([4,5])
one_step_circuit.z([4,5])
one_step_circuit.cz(4,5)
one_step_circuit.h([4,5])
one_step_circuit.draw()
```



$$G = \begin{bmatrix} \frac{2}{N} - 1 & \frac{2}{N} & \cdots & \frac{2}{N} \\ \frac{2}{N} & \frac{2}{N} - 1 & \cdots & \frac{2}{N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{N} & \frac{2}{N} & \cdots & \frac{2}{N} - 1 \end{bmatrix}.$$

$N = 4$の場合

$$G = \begin{pmatrix} -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \end{pmatrix}$$

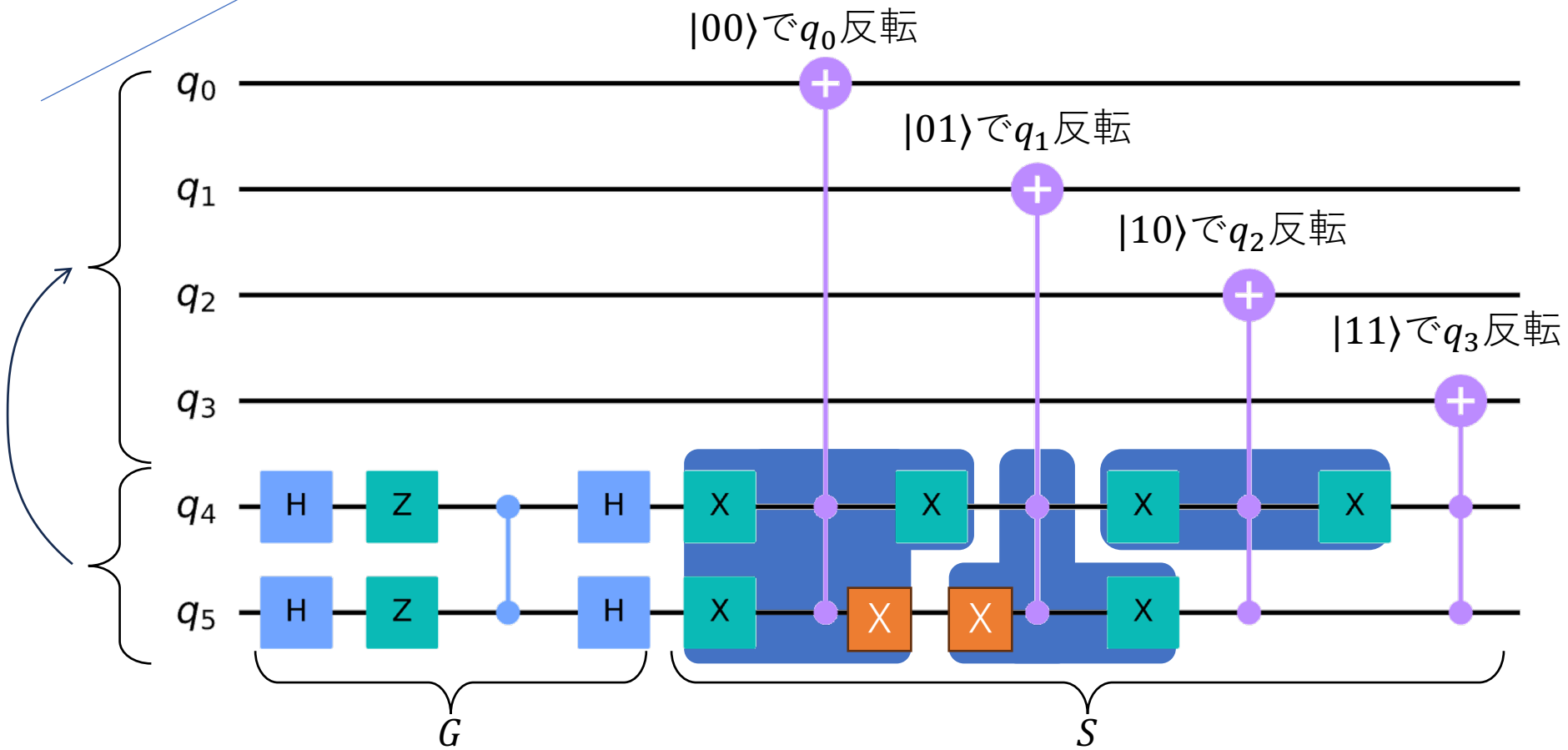
シフト演算子 S

```
# Shift operator function for 4d-hypercube
def shift_operator(circuit):
    for i in range(0,4):
        circuit.x(4)
        if i%2==0:
            circuit.x(5)
        circuit.ccx(4,5,i)
```

shift_operator(one_step_circuit)

```
one_step_gate = one_step_circuit.to_instruction()
one_step_circuit.draw()
```

q_4, q_5を制御ビットに
$q_0 \sim q_3$を反転



5. 量子ウォークによる探索アルゴリズム

グラフ上にマークした要素を見つける問題を解く。

マークした頂点集合$|M|$、遷移行列P、ユニタリ演算子$W(P)$

頂点xに隣接する頂点の一様(※)な重ね合わせ状態

(※ P_{xy}はyに依存しない)

$$|p_x\rangle = \sum_y \sqrt{P_{xy}}|y\rangle$$

$|x\rangle|y\rangle$を基底とし、 xがマークされている基底をgood(G)、それ以外をbad(B)とする。

$$|G\rangle = \frac{1}{\sqrt{|M|}} \sum_{x\in|M|} |x\rangle|p_x\rangle, \quad |B\rangle = \frac{1}{\sqrt{N-|M|}} \sum_{x\notin|M|} |x\rangle|p_x\rangle$$

$\theta = \arcsin(\sqrt{\varepsilon})$ とする。ここで、$\varepsilon = |M|/N$

アルゴリズム

1.　全ての辺の一様な重ね合わせ状態を準備

$$|U\rangle = \frac{1}{\sqrt{N}}\sum_{x}|x\rangle|p_x\rangle = \sin\theta|G\rangle + \cos\theta|B\rangle$$

2. 以下を、$\mathcal{O}(1/\sqrt{\varepsilon})$回繰り返し

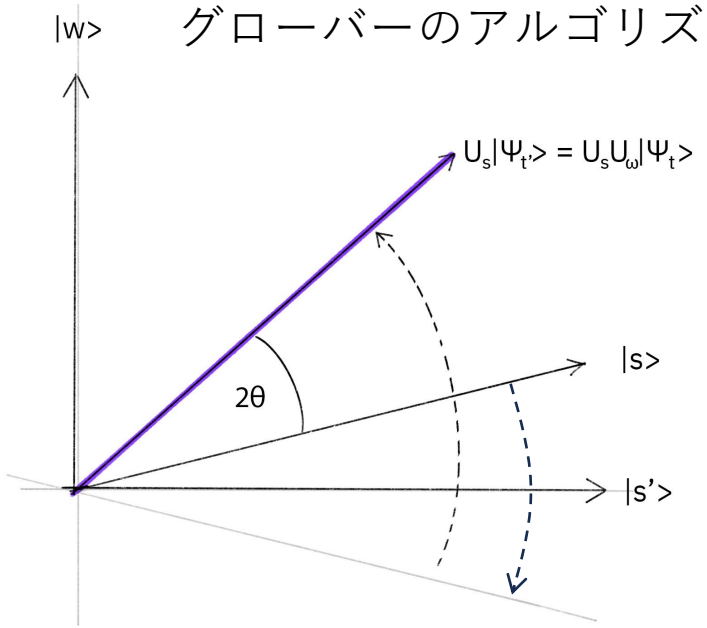
　（a）$|B\rangle$での反射　（位相オラクル[探したい状態に-1を掛ける]）

　（b）$|U\rangle$での反射　（こちらの実装は複雑）

3. 計算基底で測定

εが不明な時は、1/2, 1/4, 1/8, … と
指数的に小さくしながら再度実行する。
それでもマークされた頂点が見つからな
い時は、マークが無いと結論付ける。

グローバーのアルゴリズムと同じ



アルゴリズムの解説（文献[4]）より

$$\mathcal{A} = \mathrm{span}\{|x\rangle|p_x\rangle\}, \qquad \mathcal{B} = \mathrm{span}\{|p_y\rangle|y\rangle\}$$

とし、 $\mathcal{A}$および$\mathcal{B}$による反射を$\mathrm{ref}(\mathcal{A}), \mathrm{ref}(\mathcal{B})$とし、

$$W(P) = \mathrm{ref}(\mathcal{A})\mathrm{ref}(\mathcal{B})$$

と定義する。また、

$$D_1|x\rangle|0\rangle = |x\rangle|p_x\rangle, \qquad D_2\,|0\rangle|y\rangle = |p_y\rangle|y\rangle$$

が実装できると仮定する。

$$\mathrm{ref}(\mathcal{A}) = D_1^{\dagger}(2|0\rangle\langle 0|-I)D_1$$

$$\mathrm{ref}(\mathcal{B}) = D_2^{\dagger}(2|0\rangle\langle 0|-I)D_2$$

（D_1, D_2は部分空間において、アダマールのようなもの）

$|U\rangle$は$W(P)$の固有値$1(\theta = 0)$の固有ベクトル
（$|U\rangle$に関する反射だから？）

$|U\rangle$に対する反射を行うには、$|U\rangle$に直行する成分、つまり$\theta \neq 0$な状態全てに-1を掛ける。

入力$|w\rangle = \sum_j \alpha_j|w_j\rangle$とアンシラ$|0\rangle$を使い、
位相推定して$\theta \neq 0$に対して-1を掛ける

$$|w\rangle|0\rangle \mapsto \sum_j \alpha_j|w_j\rangle|\widetilde{\theta_j}\rangle \qquad \text{位相推定}$$

$$\mapsto \sum_j (-1)^{|\widetilde{\theta_j}\neq 0|}\alpha_j|w_j\rangle|\widetilde{\theta_j}\rangle \qquad \text{マーク}$$

$$\mapsto \sum_j (-1)^{|\widetilde{\theta_j}\neq 0|}\alpha_j|w_j\rangle|0\rangle \qquad \text{逆位相推定}$$

スペクトルギャップがδの時、 位相推定には
$\mathcal{O}(1/\sqrt{\delta})$回の実行が必要。

5. 4次元超立方体上の量子ウォークによる探索

コイン量子ウォークで実装する。

1.　頂点レジスタの全ての量子ビットにアダマールゲートを作用させて全ての辺の一様な重ね合わせ状態を作る。

2. 以下を繰り返す

(a)　位相オラクルを行う。(探したい状態に-1を掛ける)

(b)　・超立方体の量子ウォークの一ステップ分の位相推定
・ $\theta_j \neq 0$ の量子状態のマーク
・逆位相推定(uncomputing)

3. 計算基底で測定

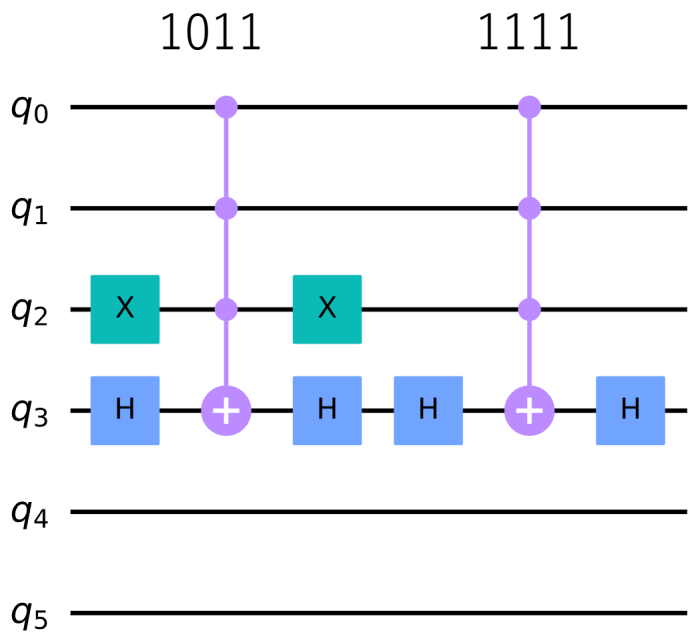
アルゴリズム
1.　全ての辺の一様な重ね合わせ状態を準備

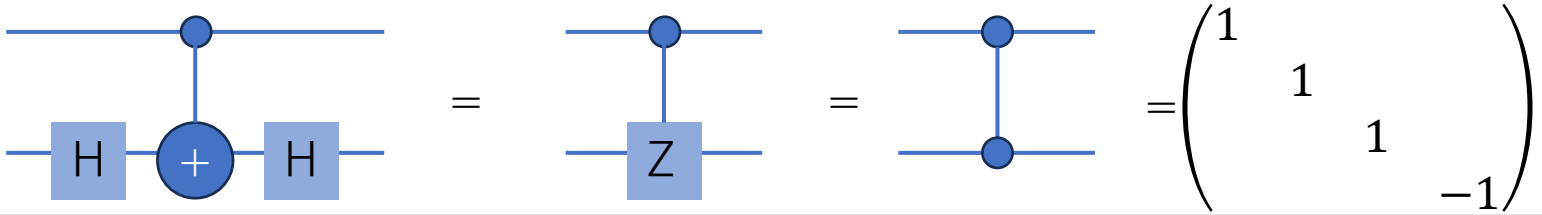
$$|U\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle|p_x\rangle = \sin\theta|G\rangle + \cos\theta|B\rangle$$

2. 以下を、$\mathcal{O}(1/\sqrt{\varepsilon})$回繰り返し
(a)　$|B\rangle$での反射　(位相オラクル[探したい状態に-1を掛ける])
(b)　$|U\rangle$での反射　(こちらの実装は複雑)
3. 計算基底で測定

アルゴリズム 2(a)
位相オラクル(探索したい状態に-1を掛ける)

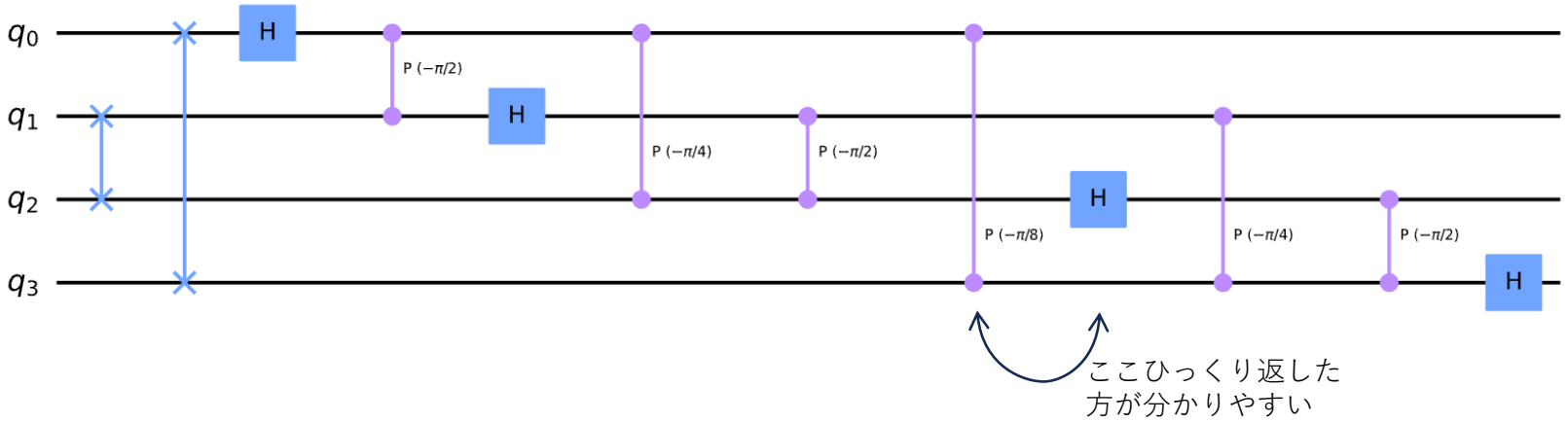
```
phase_circuit =  QuantumCircuit(6, name=' phase oracle ')
# Mark 1011
phase_circuit.x(2)
phase_circuit.h(3)
phase_circuit.mct([0,1,2], 3)
phase_circuit.h(3)
phase_circuit.x(2)
# Mark 1111
phase_circuit.h(3)
phase_circuit.mct([0,1,2],3)
phase_circuit.h(3)
phase_oracle_gate = phase_circuit.to_instruction()
# Phase oracle circuit
phase_oracle_circuit =  QuantumCircuit(11, name=' PHASE ORACLE CIRCUIT ')
phase_oracle_circuit.append(phase_oracle_gate, [4,5,6,7,8,9])
phase_circuit.draw()
```



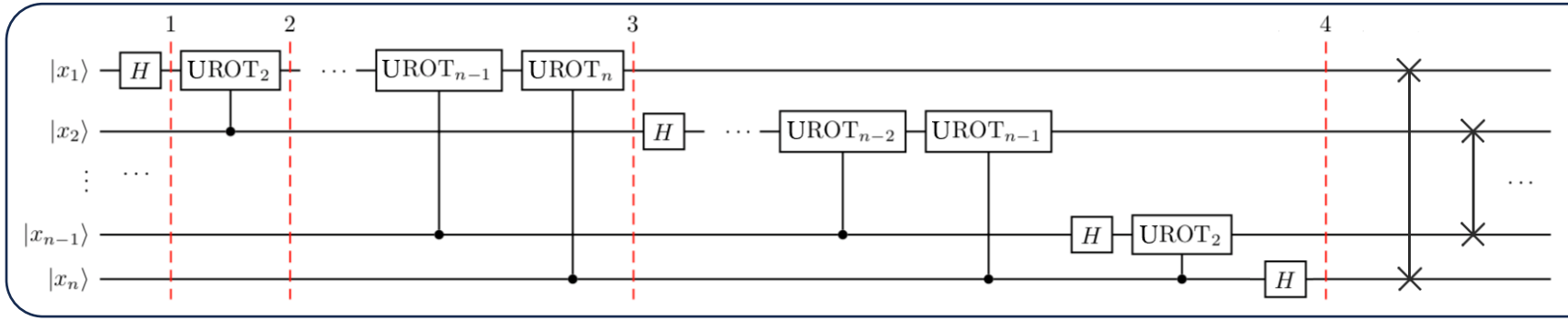


位相推定のための量子フーリエ変換と逆量子フーリエ変換

```
inv_qft_gate = QFT(4, inverse=True).to_instruction()
qft_gate = QFT(4, inverse=False).to_instruction()

QFT(4, inverse=True).decompose().draw("mpl")
```



4量子ビットの
逆量子フーリエ変換

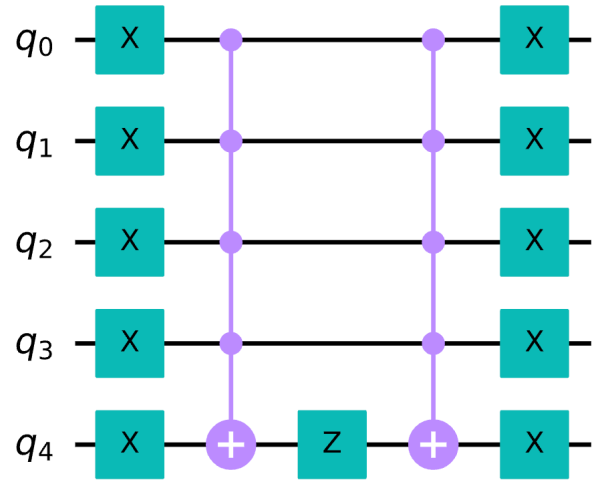


4量子ビットの
量子フーリエ変換
(Qiskit 量子フーリエ変換より)

アルゴリズム 2(b) - 2

位相オラクル($\theta_j \neq 0$ の状態に-1を掛ける)

```
# Mark q_4 if the other qubits are non-zero
mark_auxiliary_circuit = QuantumCircuit(5, name=' mark
auxiliary ')
mark_auxiliary_circuit.x([0,1,2,3,4])
mark_auxiliary_circuit.mct([0,1,2,3], 4)
mark_auxiliary_circuit.z(4)
mark_auxiliary_circuit.mct([0,1,2,3], 4)
mark_auxiliary_circuit.x([0,1,2,3,4])

mark_auxiliary_gate = mark_auxiliary_circuit.to_instruction()
mark_auxiliary_circuit.draw()
```



マークなし: $q_0q_1q_2q_3 = 0$の時 q_4 : XXZXX = Z $= \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$

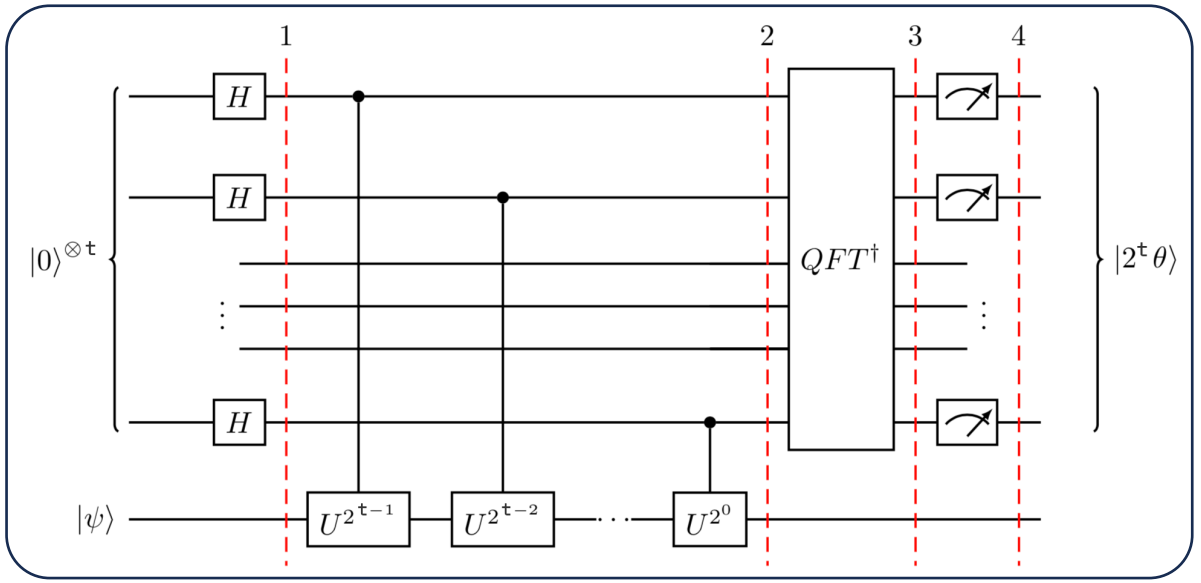
マークあり: $q_0q_1q_2q_3 = 1$の時 q_4 : XZX $= \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$

$a|0\rangle + b|1\rangle \mapsto \begin{cases} a|0\rangle - b|1\rangle \\ -a|0\rangle + b|1\rangle \end{cases}$ となるため、-1が掛かっているのと等価。

(もう一度 Z を掛ければ、マークなしはI、マークありは$-I$となる。)

アルゴリズム 2(b) 全体
位相推定 + $\theta_j \neq 0$ の量子状態のマーク + 逆位相推定

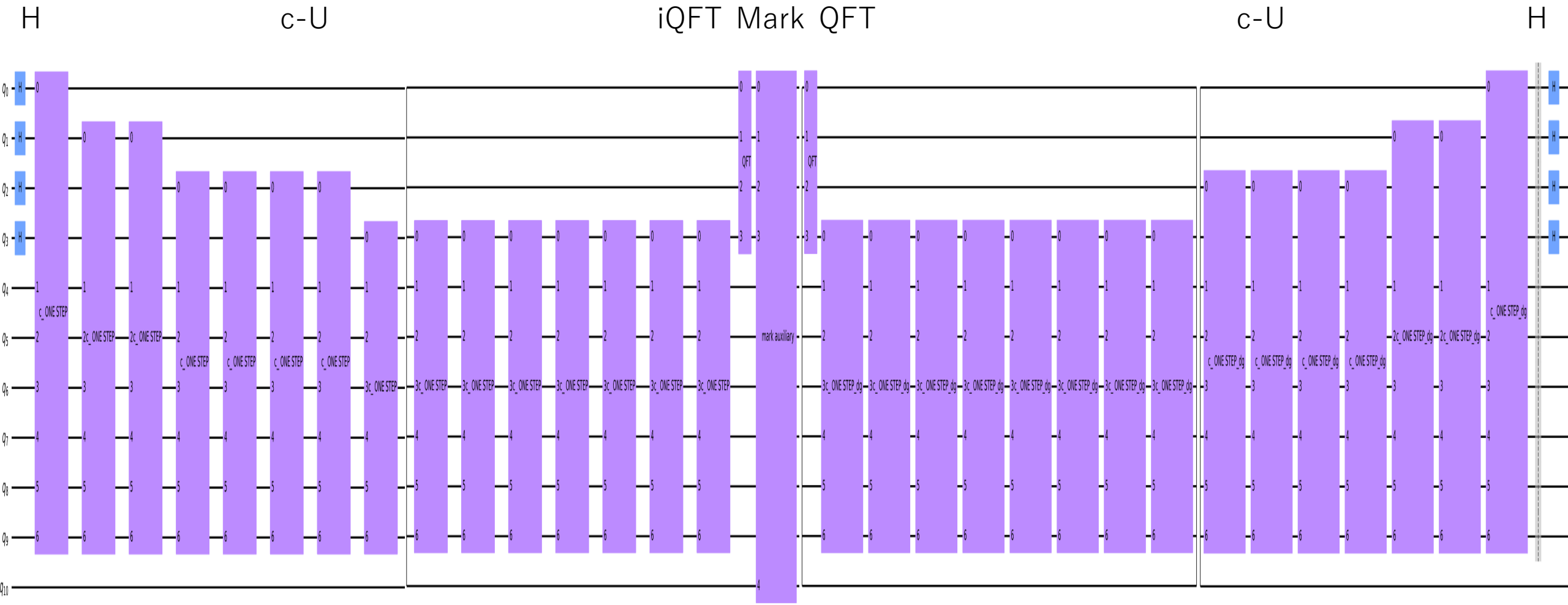
```
# Phase estimation
phase_estimation_circuit = QuantumCircuit(11, name=' phase estimation ')
phase_estimation_circuit.h([0,1,2,3])
for i in range(0,4):
    stop = 2**i
    for j in range(0,stop):
        phase_estimation_circuit.append(cont_one_step, [i,4,5,6,7,8,9])

# Inverse fourier transform
phase_estimation_circuit.append(inv_qft_gate, [0,1,2,3])

# Mark all angles theta that are not 0 with an auxiliary qubit
phase_estimation_circuit.append(mark_auxiliary_gate, [0,1,2,3,10])

# Reverse phase estimation
phase_estimation_circuit.append(qft_gate, [0,1,2,3])

for i in range(3,-1,-1):
    stop = 2**i
    for j in range(0,stop):
        phase_estimation_circuit.append(inv_cont_one_step, [i,4,5,6,7,8,9])
phase_estimation_circuit.barrier(range(0,10))
phase_estimation_circuit.h([0,1,2,3])

# Make phase estimation gate
phase_estimation_gate = phase_estimation_circuit.to_instruction()
phase_estimation_circuit.draw()
```



… 8 4 2 1

↑ べき乗のため、
同じ回路を何度も実行する

アルゴリズム 2(b) 全体


H
c-U
iQFT
Mark
QFT
c-U
H
c_ONE STEP
2c_ONE STEP
3c_ONE STEP
mark auxiliary
QFT
c_ONE STEP_dg
2c_ONE STEP_dg
3c_ONE STEP_dg

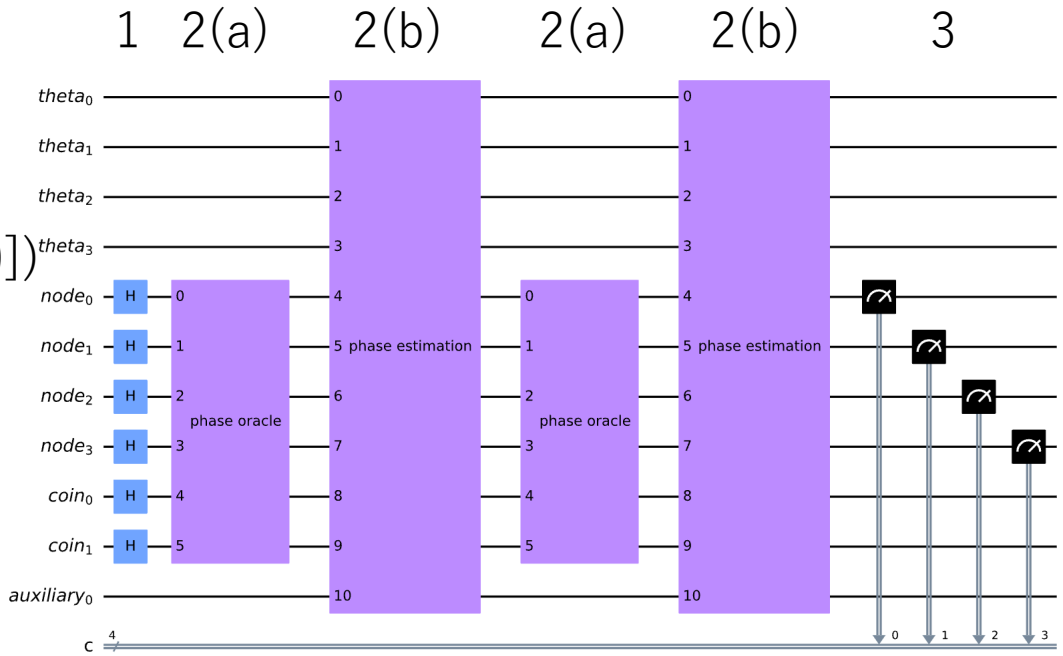
アルゴリズム全体
全回路

```
# Implementation of the full quantum walk search algorithm
theta_q = QuantumRegister(4, 'theta')
node_q = QuantumRegister(4, 'node')
coin_q = QuantumRegister(2, 'coin')
auxiliary_q = QuantumRegister(1, 'auxiliary')
creg_c2 = ClassicalRegister(4, 'c')
circuit = QuantumCircuit(theta_q, node_q, coin_q, auxiliary_q, creg_c2)
# Apply Hadamard gates to the qubits that represent the nodes and the coin
circuit.h([4,5,6,7,8,9])
iterations = 2

for i in range(0,iterations):
    circuit.append(phase_oracle_gate, [4,5,6,7,8,9])
    circuit.append(phase_estimation_gate, [0,1,2,3,4,5,6,7,8,9,10])

circuit.measure(node_q[0], creg_c2[0])
circuit.measure(node_q[1], creg_c2[1])
circuit.measure(node_q[2], creg_c2[2])
circuit.measure(node_q[3], creg_c2[3])
circuit.draw()
```

アルゴリズム
1. 全ての辺の一様な重ね合わせ状態を準備

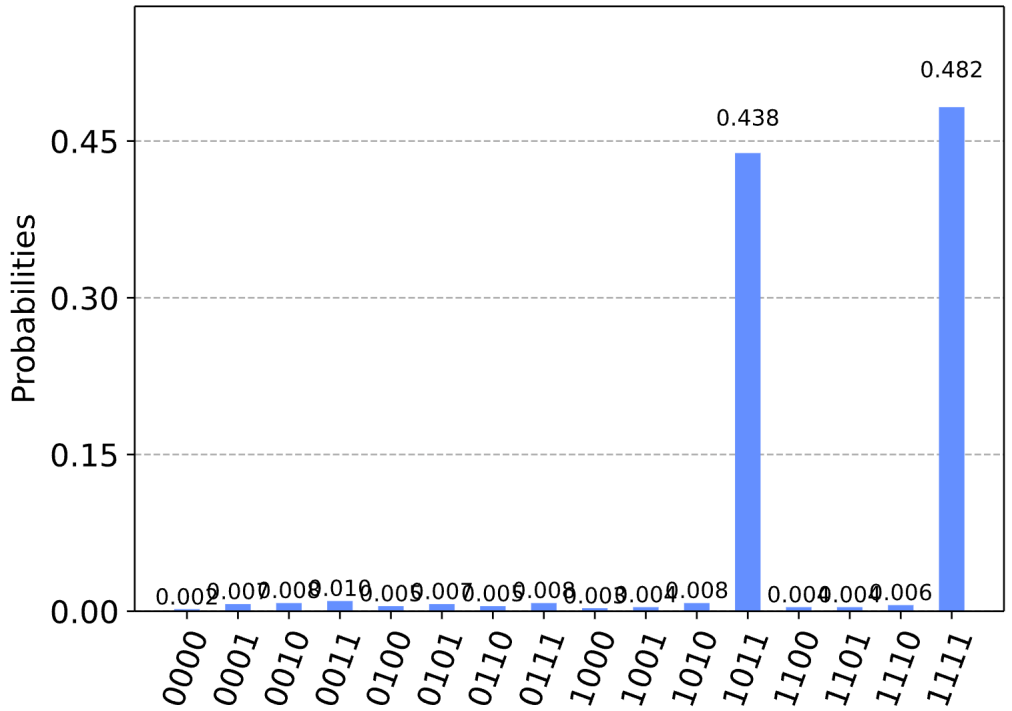
$$|U\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle|p_x\rangle = \sin\theta|G\rangle + \cos\theta|B\rangle$$

2. 以下を、$\mathcal{O}(1/\sqrt{\varepsilon})$回繰り返し
   (a) $|B\rangle$での反射 (位相オラクル[探したい状態に-1を掛ける])
   (b) $|U\rangle$での反射 (こちらの実装は複雑)
3. 計算基底で測定

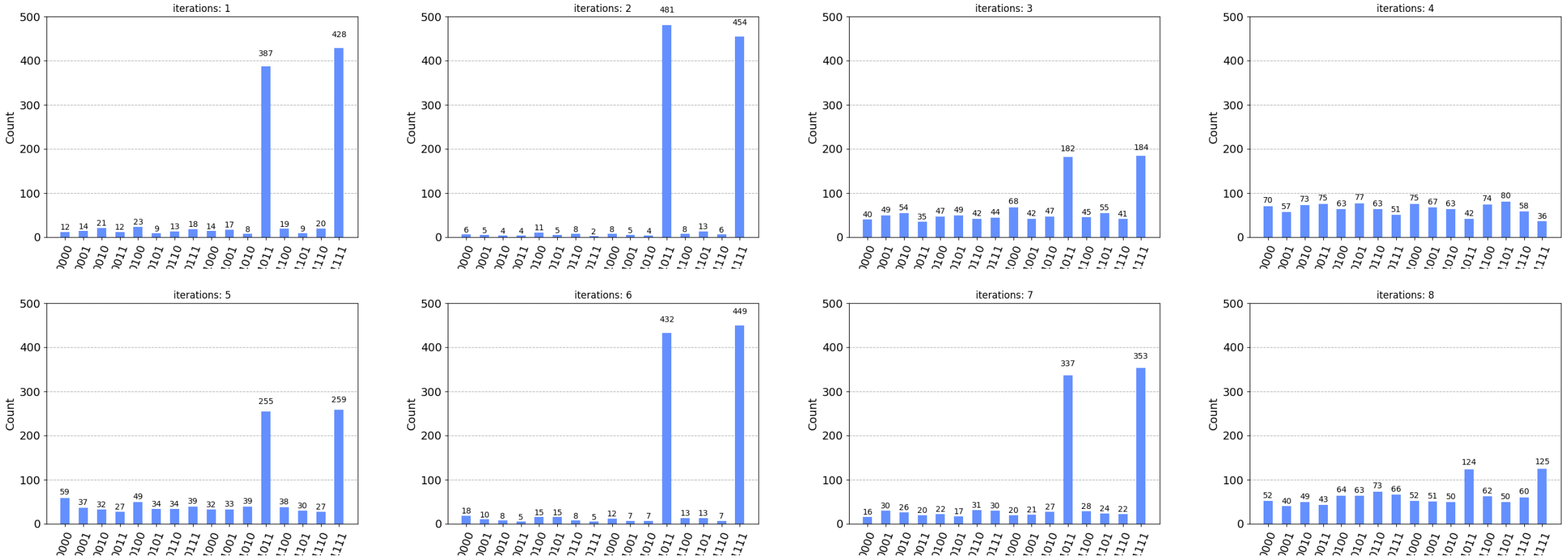


シミュレーション

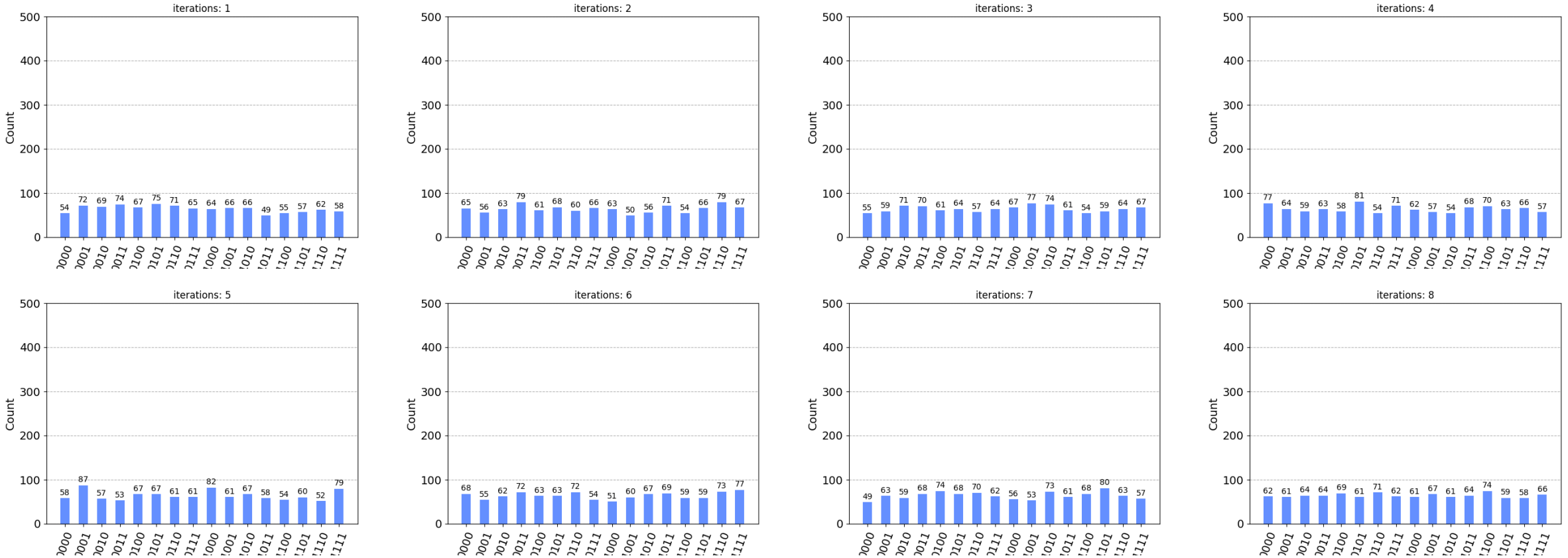
```
backend = Aer.get_backend('qasm_simulator')
job = execute( circuit, backend, shots=1024 )
hist = job.result().get_counts()
plot_histogram( hist )
```



実験: 繰り返し回数を変更



実験: マークなし、繰り返し回数を変更



参考文献

1. Renato Portugal. Quantum Walks and Search Algorithms. New York, NY: Springer New York, 2013 量子ウォークの教科書

2. Markus G. Kuhn.Some Introductory Notes on Quantum Computing. Apr. 2000 多分直接関係ない？

3. Thomas G. Wong. “Equivalence of Szegedy’s and coined quantum walks”. In: Quantum InformationProcessing 16.9 (July 2017). ISSN: 1573-1332. DOI:10.1007/s11128-017-1667-y. URL:http://dx.doi.org/10.1007/s11128-017-1667-y.37コイン量子ウォークとセゲディ量子ウォークの等価性

★ 4. Ronald de Wolf. Quantum Computing: Lecture Notes. 2021. arXiv:1907.09415 [quant-ph] 量子ウォークによる探索アルゴリズム

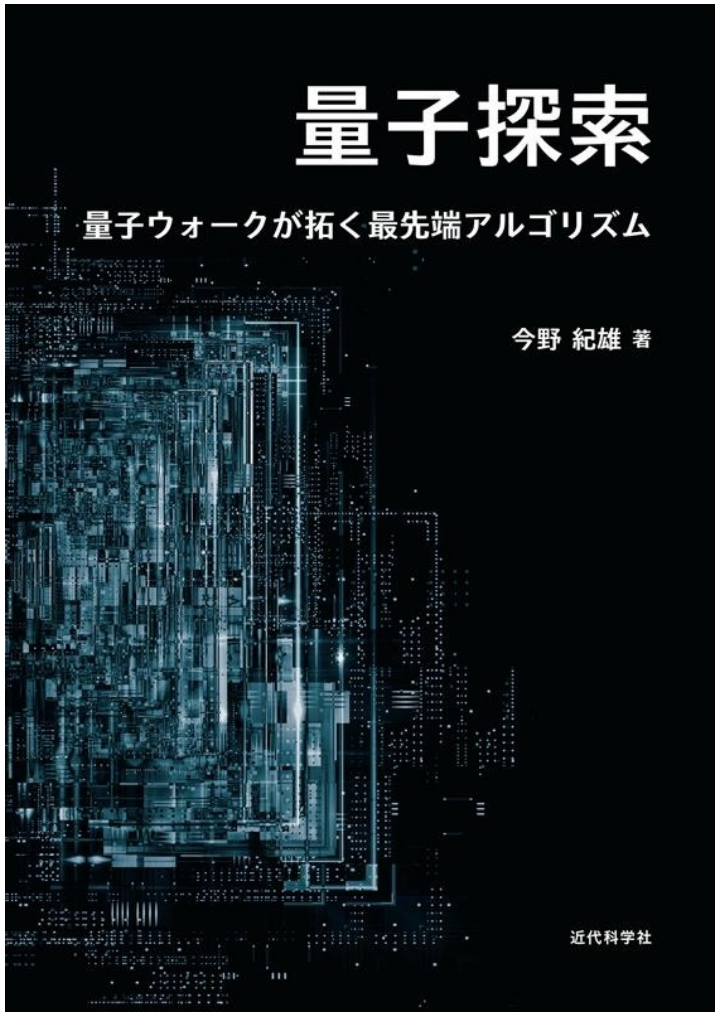
★ arXiv:1907.09415

Quantum Computing: Lecture Notes

Ronald de Wolf

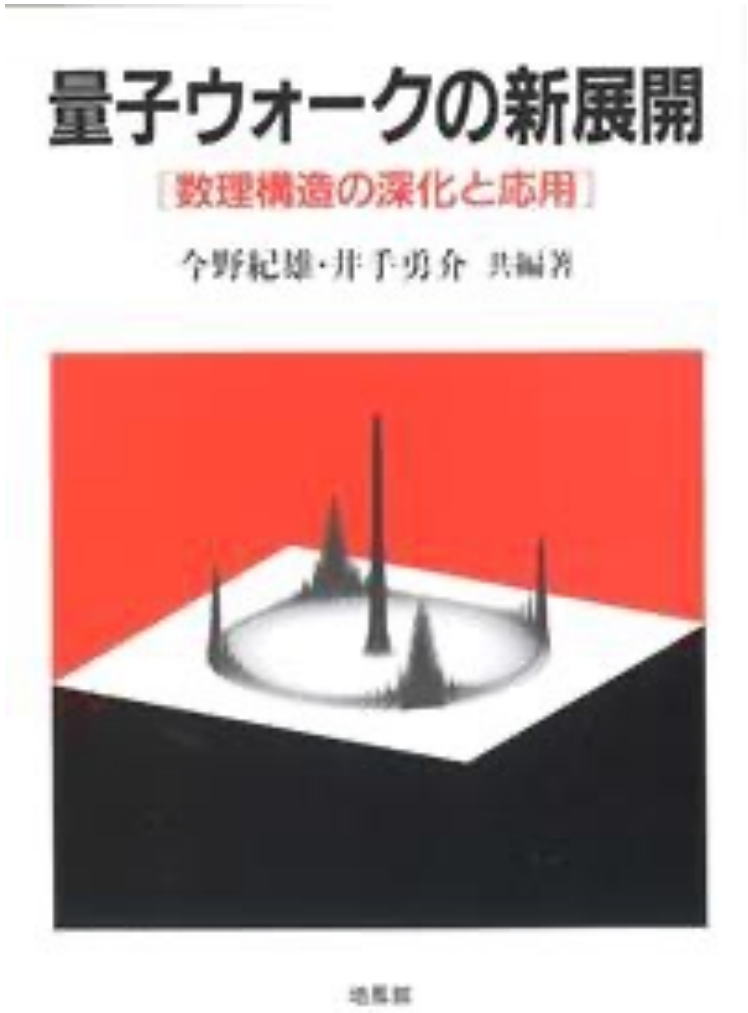
QuSoft, CWI and University of Amsterdam

チュートリアル 4章はここの内容



日本語のテキストの中では
チュートリアルに一番近い内容

第1章 グローヴァー・アルゴリズム
第2章 サイクル上の量子ウォーク
第3章 超立方格子と区間の量子ウォーク
第4章 2次元トーラス上の量子ウォーク
第5章 空間的な量子探索アルゴリズム
第6章 無向2部グラフ上の量子ウォーク
第7章 有向2部グラフ上の量子探索



応用事例が豊富

12. 量子ウォーク同位体分離
13. 量子計算シミュレーションに向けて：
    光学と量子ウォーク
14. 量子ウォークの物理的実装方法
15. トポロジカル絶縁体と量子ウォーク
16. 量子ウォークを用いた金融の時系列解析