量子分散計算

ルガル　フランソワ

京都大学
情報学研究科

量子情報·物性の新潮流
2018年8月2日

自己紹介

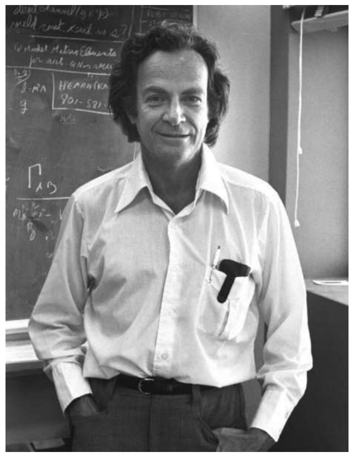
- 1978年　フランス生まれ
- 2000年　リヨン工科大学（フランス）　卒業
- 2000年　来日

- 2001年～2003年　東京大学大学院新領域創成科学研究科　修士課程
  研究テーマ：ニューラルネットワーク、機械学習
- 2003年～2006年　東京大学大学院情報理工学系研究科　　博士課程
  研究テーマ：量子計算

- 2006年～2009年 JST ERATO-SORST 量子システムアーキテクチャ 研究員
  研究テーマ：量子アルゴリズム、量子計算量理論
- 2009年～2016年 東京大学大学院情報理工学系研究科　特任講師➡特任准教授
  研究テーマ：量子アルゴリズム、行列積アルゴリズム
- 2016年～　　　京都大学大学院情報学研究科　　　特定准教授
  研究テーマ：量子計算、分散計算

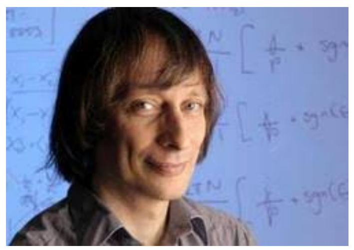
量子計算の歴史

量子計算の提唱



Feynman

1982



Deutsch

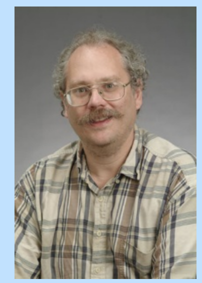
1985

量子論理演算の実現

Wineland Haroche

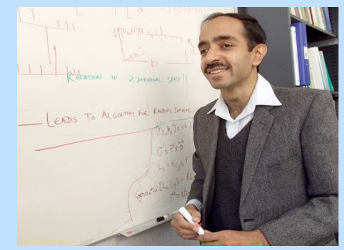
1次量子計算ブーム

高速な
量子アルゴリズムの発明



Shor

1994



Grover

1996

量子誤り訂正

1999

Ambainis

2005

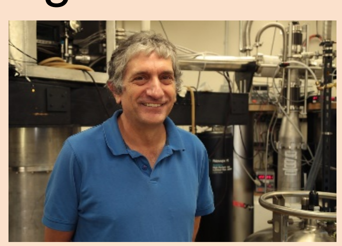
2次量子計算ブーム

小規模量子コンピュータの構築
(20~100量子ビット)

D-Wave

2011

Google & Martinis



2014

IBM

2017

Microsoft,
Rigetti,...

量子アルゴリズム

~~これだけ?~~

- quantum algorithms with amplitude amplification [Brassard+ 1999]
- quantum algorithms for element disjointness [Ambainis 2002]
- quantum algorithms for Gauss sums [van Dam + 2002]
- quantum algorithms for solving Pell's equation [Hallgren 2002]
- quantum algorithms for quantum simulations [Childs 2004]
- quantum algorithms for hidden subgroups [Kuperberg 2004]
- quantum algorithms for finding an unit group [Hallgren 2005]
- quantum algorithms for triangle finding [Magniez+ 2005]
- quantum algorithms for computing knot invariants [Aharonov+ 2006]
- quantum algorithms for data streams [Le Gall 2006]
- quantum algorithms for hidden nonlinear structures [Childs+ 2007]
- quantum algorithms for linear equations [Harrow+ 2009]
- quantum algorithms for group isomorphism [Le Gall 2010]
- quantum algorithms for matrix multiplication [Le Gall 2011]
- quantum algorithms using span programs [Belovs 2011]
- quantum algorithms for matrix inversion [Ta-Shma 2013]
- quantum algorithms for combinatorial triangle finding [Le Gall 2011]
- quantum algorithms for pattern matching [Montanaro 2014]
- quantum algorithms for distributed computation [Le Gall and Magniez 2017]
- …

Shor's algorithm (1994)
素因数分解

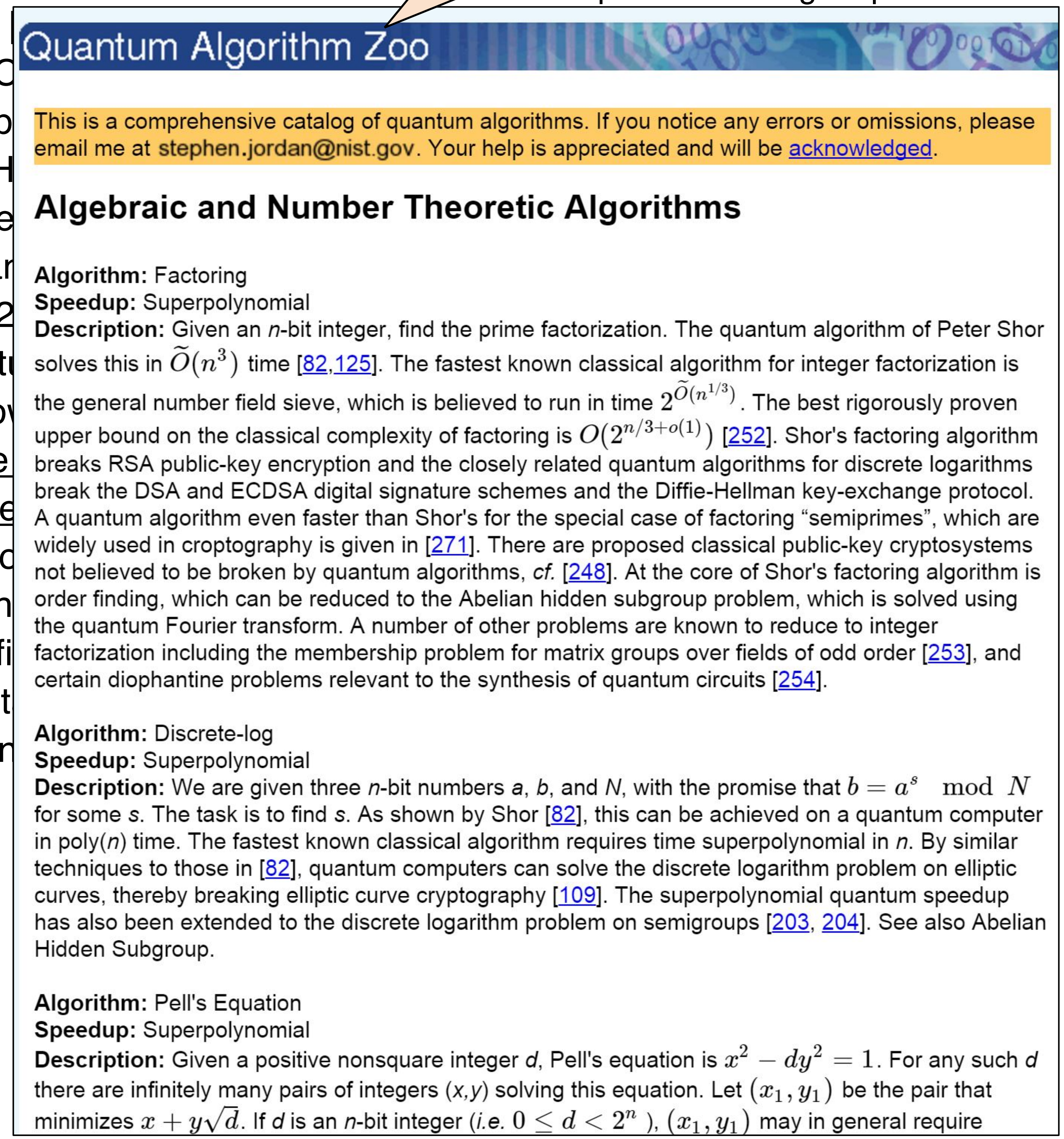
Grover's algorithm (1996)
量子探索

量子アルゴリズム

- quantum algorithms with amplitude amplification [Brassard+ 1999]
- quantum algorithms for element disjointness [Ambainis 2002]
- quantum algorithms for Gauss sums [van Dam + 2002]
- quantum algorithms for solving Pell's equation [
- quantum algorithms for quantum simulations [C
- quantum algorithms for hidden subgroups [Kup
- quantum algorithms for finding an unit group [H
- quantum algorithms for triangle finding [Magnie
- quantum algorithms for computing knot invariar
- quantum algorithms for data streams [Le Gall 2
- quantum algorithms for hidden nonlinear structu
- quantum algorithms for linear equations [Harrov
- quantum algorithms for group isomorphism [Le
- quantum algorithms for matrix multiplication [Le
- quantum algorithms using span programs [Belo
- quantum algorithms for matrix inversion [Ta-Sh
- quantum algorithms for combinatorial triangle fi
- quantum algorithms for pattern matching [Mont
- quantum algorithms for distributed computation
- …

量子アルゴリズム園

https://math.nist.gov/quantum/zoo/


Quantum Algorithm Zoo

This is a comprehensive catalog of quantum algorithms. If you notice any errors or omissions, please email me at stephen.jordan@nist.gov. Your help is appreciated and will be acknowledged.

Algebraic and Number Theoretic Algorithms

Algorithm: Factoring
Speedup: Superpolynomial
Description: Given an *n*-bit integer, find the prime factorization. The quantum algorithm of Peter Shor solves this in $\widetilde{O}(n^3)$ time [82,125]. The fastest known classical algorithm for integer factorization is the general number field sieve, which is believed to run in time $2^{\widetilde{O}(n^{1/3})}$. The best rigorously proven upper bound on the classical complexity of factoring is $O(2^{n/3+o(1)})$ [252]. Shor's factoring algorithm breaks RSA public-key encryption and the closely related quantum algorithms for discrete logarithms break the DSA and ECDSA digital signature schemes and the Diffie-Hellman key-exchange protocol. A quantum algorithm even faster than Shor's for the special case of factoring "semiprimes", which are widely used in croptography is given in [271]. There are proposed classical public-key cryptosystems not believed to be broken by quantum algorithms, *cf.* [248]. At the core of Shor's factoring algorithm is order finding, which can be reduced to the Abelian hidden subgroup problem, which is solved using the quantum Fourier transform. A number of other problems are known to reduce to integer factorization including the membership problem for matrix groups over fields of odd order [253], and certain diophantine problems relevant to the synthesis of quantum circuits [254].

Algorithm: Discrete-log
Speedup: Superpolynomial
Description: We are given three *n*-bit numbers *a*, *b*, and *N*, with the promise that $b = a^s \mod N$ for some *s*. The task is to find *s*. As shown by Shor [82], this can be achieved on a quantum computer in poly(*n*) time. The fastest known classical algorithm requires time superpolynomial in *n*. By similar techniques to those in [82], quantum computers can solve the discrete logarithm problem on elliptic curves, thereby breaking elliptic curve cryptography [109]. The superpolynomial quantum speedup has also been extended to the discrete logarithm problem on semigroups [203, 204]. See also Abelian Hidden Subgroup.

Algorithm: Pell's Equation
Speedup: Superpolynomial
Description: Given a positive nonsquare integer *d*, Pell's equation is $x^2 - dy^2 = 1$. For any such *d* there are infinitely many pairs of integers (*x,y*) solving this equation. Let (x_1, y_1) be the pair that minimizes $x + y\sqrt{d}$. If *d* is an *n*-bit integer (*i.e.* $0 \leq d < 2^n$), (x_1, y_1) may in general require

278 件 (2018年7月時点)

量子アルゴリズム

- quantum algorithms with amplitude amplification [Brassard+ 1999]
- quantum algorithms for element disjointness [Ambainis 2002]
- quantum algorithms for Gauss sums [van Dam + 2002]
- quantum algorithms for solving Pell's equation [Hallgren 2002]
- quantum algorithms for quantum simulations [Childs 2004]
- quantum algorithms for hidden subgroups [Kuperberg 2004]
- quantum algorithms for finding an unit group [Hallgren 2005]
- quantum algorithms for triangle finding [Magniez+ 2005]
- quantum algorithms for computing knot invariants [Aharonov+ 2006]
- quantum algorithms for data streams [Le Gall 2006]
- quantum algorithms for hidden nonlinear structures [Childs+ 2007]
- quantum algorithms for linear equations [Harrow+ 2009]
- quantum algorithms for group isomorphism [Le Gall 2010]
- quantum algorithms for matrix multiplication [Le Gall 2011]
- quantum algorithms using span programs [Belovs 2011]
- quantum algorithms for matrix inversion [Ta-Shma 2013]
- quantum algorithms for combinatorial triangle finding [Le Gall 2011]
- quantum algorithms for pattern matching [Montanaro 2014]
- quantum algorithms for distributed computation [Le Gall and Magniez 2017]
- …

行列積に対する量子アルゴリズム

疎行列に対して古典計算より速い

量子アルゴリズム

- quantum algorithms with amplitude amplification [Brassard+ 1999]
- quantum algorithms for element disjointness [Ambainis 2002]
- quantum algorithms for Gauss sums [van Dam + 2002]
- quantum algorithms for solving Pell's equation [Hallgren 2002]
- quantum algorithms for quantum simulations [Childs 2004]
- quantum algorithms for hidden subgroups [Kuperberg 2004]
- quantum algorithms for finding an unit group [Hallgren 2005]
- quantum algorithms for triangle finding [Magniez+ 2005]
- quantum algorithms for computing knot invariants [Aharonov+ 2006]
- quantum algorithms for data streams [Le Gall 2006]
- quantum algorithms for hidden nonlinear structures [Childs+ 2007]
- quantum algorithms for linear equations [Harrow+ 2009]
- quantum algorithms for group isomorphism [Le Gall 2010]
- quantum algorithms for matrix multiplication [Le Gall 2011]
- quantum algorithms using span programs [Belovs 2011]
- quantum algorithms for matrix inversion [Ta-Shma 2013]
- quantum algorithms for combinatorial triangle finding [Le Gall 2011]
- quantum algorithms for pattern matching [Montanaro 2014]
- quantum algorithms for distributed computation [Le Gall and Magniez 2017]
- …

行列の固有値を高速に
求める量子アルゴリズム

量子機械学習へ応用の可能性も

行列積に対する量子アルゴリズム

疎行列に対して古典計算より速い

量子アルゴリズム

- quantum algorithms with amplitude amplification [Brassard+ 1999]
- quantum algorithms for element disjointness [Ambainis 2002]
- quantum algorithms for Gauss sums [van Dam + 2002]
- quantum algorithms for solving Pell's equation [Hallgren 2002]
- quantum algorithms for quantum simulations [Childs 2004]
- quantum algorithms for hidden subgroups [Kuperberg 2004]
- quantum algorithms for finding an unit group [Hallgren 2005]
- quantum algorithms for triangle finding [Magniez+ 2005]
- quantum algorithms for computing knot invariants [Aharonov+ 2006]
- quantum algorithms for data streams [Le Gall 2006]
- quantum algorithms for hidden nonlinear structures [Childs+ 2007]
- quantum algorithms for linear equations [Harrow+ 2009]
- quantum algorithms for group isomorphism [Le Gall 2010]
- quantum algorithms for matrix multiplication [Le Gall 2011]
- quantum algorithms using span programs [Belovs 2011]
- quantum algorithms for matrix inversion [Ta-Shma 2013]
- quantum algorithms for combinatorial triangle finding [Le Gall 2011]
- quantum algorithms for pattern matching [Montanaro 2014]
- quantum algorithms for distributed computation [Le Gall and Magniez 2017]
- …

量子系のシミュレーション

量子化学、創薬へ応用?

行列の固有値を高速に
求める量子アルゴリズム

量子機械学習へ応用の可能性も

行列積に対する量子アルゴリズム

ほとんどは大規模量子コンピュータ向けのアルゴリズム

小規模量子コンピュータのできるもの

現在のコンピュータより速く

- ✓ 量子アニーリング
- ✓ 量子系のシミュレーション

⇨ 量子化学、創薬への応用を目指す...

高い精度が求まれて、まだ難しい...

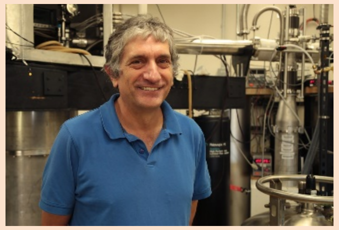
- ✓ とりあえず小規模量子コンピュータの優位性を示そう：

小規模量子コンピュータで簡単に解けて、
古典計算では解けない人工的な問題を作ろう

小規模量子コンピュータの構築
(20~100量子ビット)

IBM

Google & Martinis



QUANTUM SUPREMACY
（Boson samplingなど）

2017 2014 2005

Microsoft,
Rigetti,...

2次量子計算ブーム

量子アルゴリズムの理論研究の潮流

引き続き、大規模量子コンピュータ向けの量子アルゴリズムの開発

量子分散計算の枠組みで研究

✓ 重要な計算問題を対象とする

✓ 実装のし易さを特に求めない

QUANTUM SUPREMACY
(小〜中規模量子コンピュータの優位性の確立)

量子分散計算を用いて研究

✓ 人工的な計算問題でも良い

✓ 実装のしやすさを重視

進行中のプロジェクト：量子分散計算 (クラウド量子計算)

[Izumi and Le Gall 2017]
[Le Gall and Magniez 2018]
[Le Gall, Nishimura and Rosmanis 2018]

量子アルゴリズムの理論研究の潮流

引き続き、大規模量子コンピュータ向けの量子アルゴリズムの開発

量子分散計算の枠組みで研究

F. Le Gall and F. Magniez. **Sublinear-Time Quantum Computation of the Diameter in Distributed Networks**. Proceedings of the 37th ACM Symposium on Principles of Distributed Computing (PODC 2018).
Also arXiv: 1804.02917.

分散計算の重要な問題に対して、
古典分散アルゴリズムより高速な量子分散アルゴリズムを初めて与えた

Outline: Quantum Distributed Computing

- ✓ From the perspective of <u>computability and computational complexity</u>, quantum distributed computing has mostly been studied in the framework of 2-party communication complexity

- ✓ Relatively few results focusing on more than two parties:

  - ➢ exact quantum protocols for leader election on anonymous networks [Tani, Kobayashi, Matsumoto 2007]

  - ➢ study of quantum distributed algorithms on non-anonymous networks

    [Gavoille, Kosowski, Markiewicz 2009]
    [Elkin, Klauck, Nanongkai, Pandurangan 2014]

    <u>negative results</u>: show impossibility of quantum distributed computing faster than classical distributed computing for many important problems (shortest paths, minimum spanning tree,…)

Question: can quantum distributed computing be useful?
(on non-anonymous networks)

Our result: Yes, we can compute the diameter of the network faster!

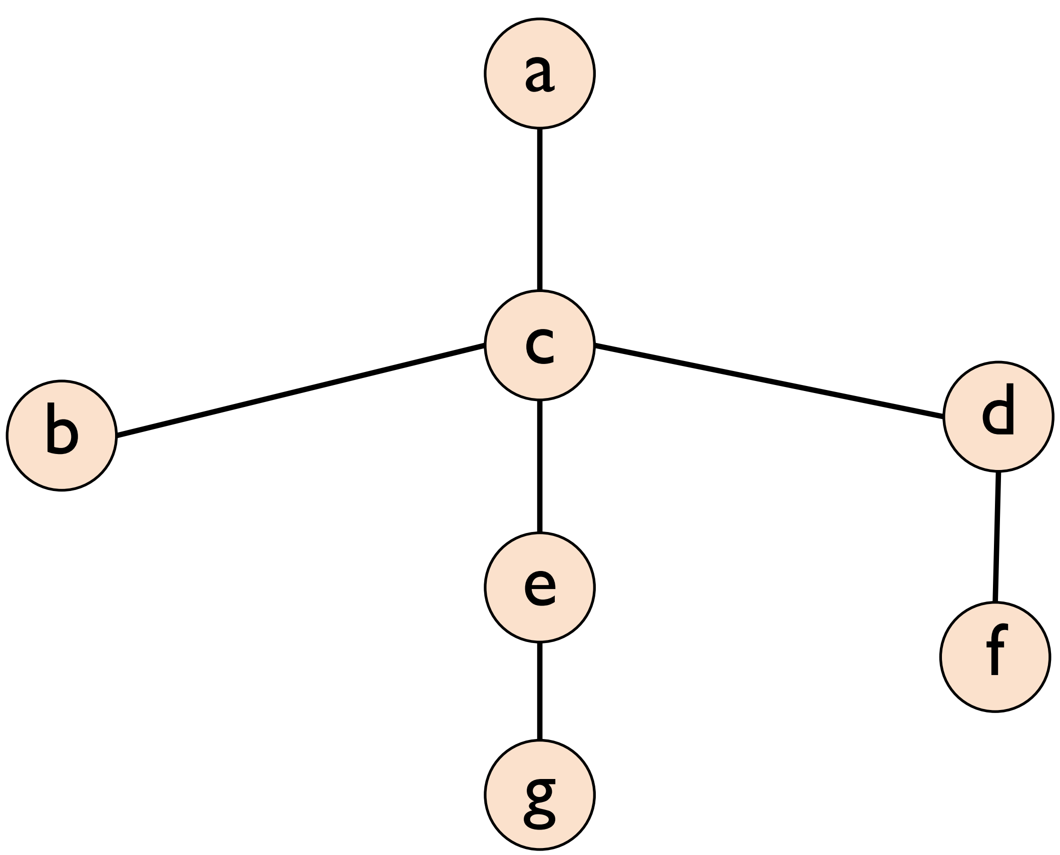
Eccentricity and Diameter

Consider an undirected and unweighted graph G = (V,E) with n nodes

The <u>diameter</u> of the graph is the maximum distance between two nodes (直径)

$$D = \max_{u,v \in V} \{d(u,v)\}$$

d(u,v) = distance between u and v



Eccentricity and Diameter

Consider an undirected and unweighted graph G = (V,E) with n nodes

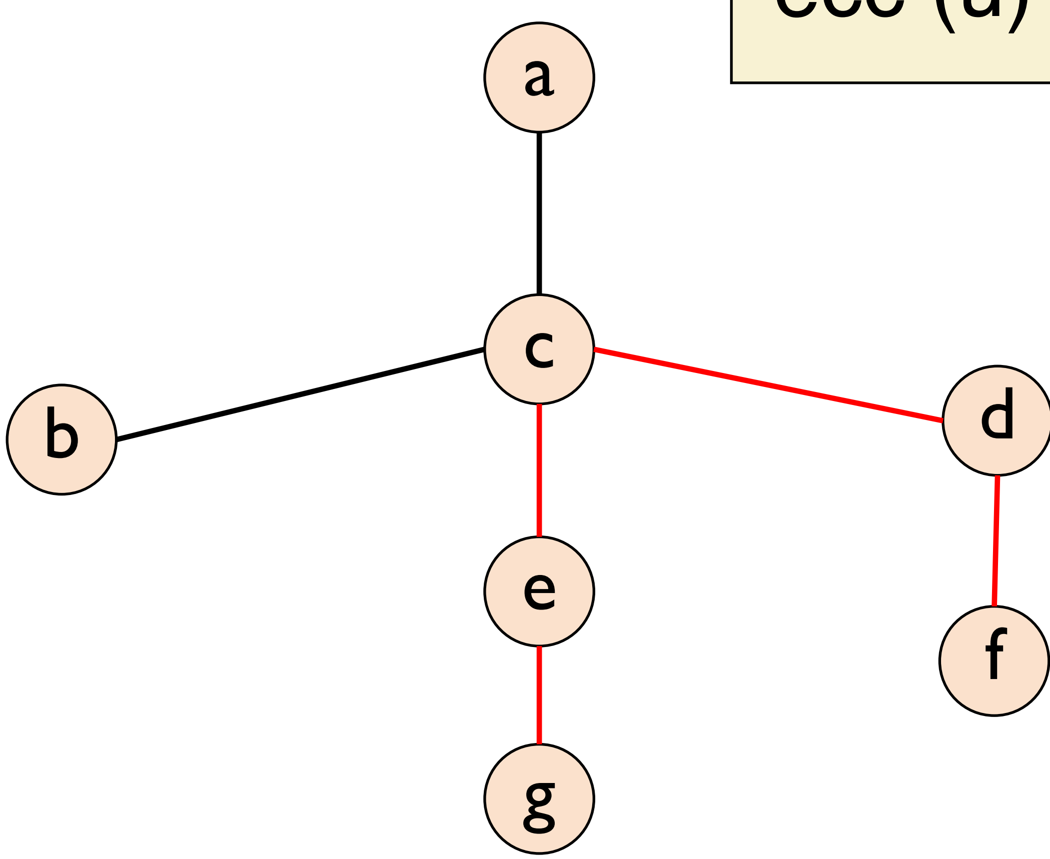
The <u>diameter</u> of the graph is the maximum distance between two nodes
(直径)

$$D = \max_{u,v \in V} \{d(u,v)\} = \max_{u \in V} \{ecc\ (u)\}$$

d(u,v) = distance between u and v

The <u>eccentricity</u> of a node u is defined as
(離心数)

$$ecc\ (u) = \max_{v \in V} \{d(u,v)\}$$



ecc (a) = 3

$$\begin{cases} d(a,a) = 0 \\ d(a,b) = 2 \\ d(a,c) = 1 \\ d(a,d) = 2 \\ d(a,e) = 2 \\ d(a,f) = 3 \\ d(a,g) = 3 \end{cases}$$

ecc (b) = 3
ecc (c) = 2
ecc (d) = 3
ecc (e) = 3
ecc (f) = 4
ecc (g) = 4

D = 4

Classical Distributed Computation of the Eccentricity

Let's write n = number of nodes

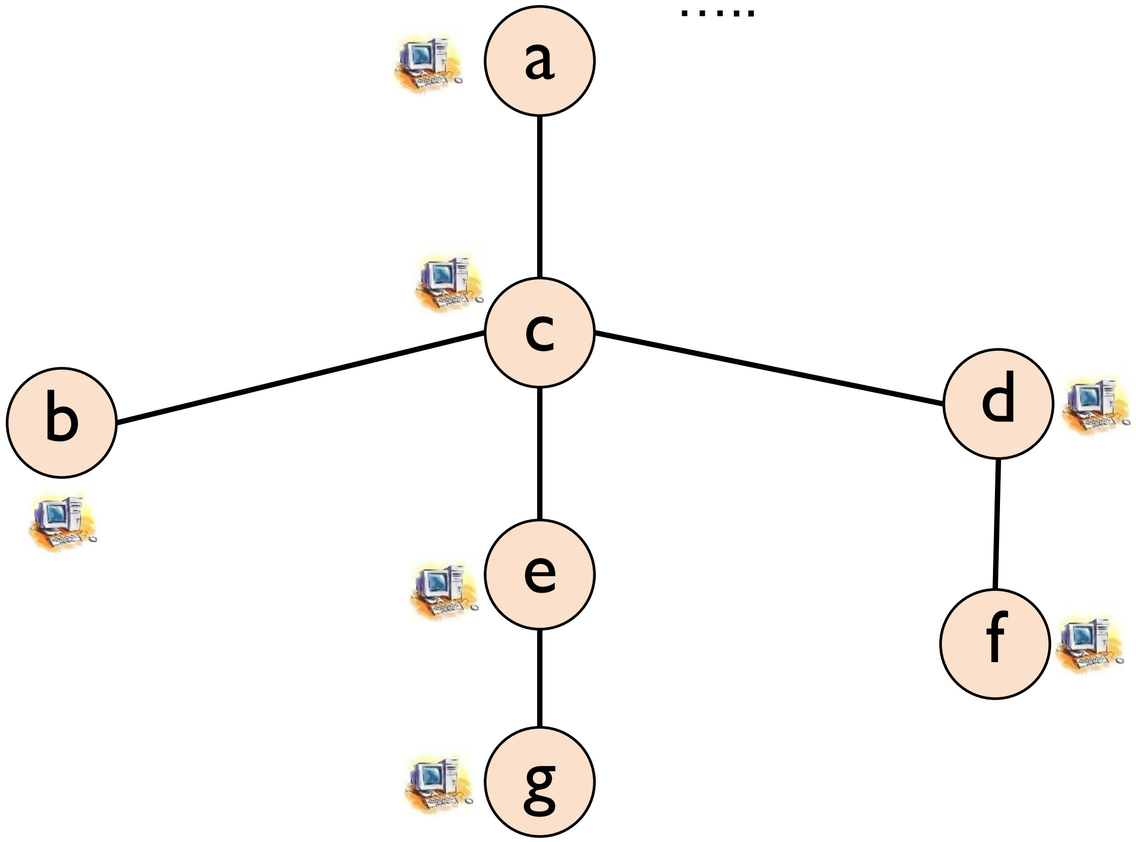
- ✓ Each node represents a processor (with a unique ID)
- ✓ Each edge represents a classical channel
- ✓ At each round only one short (i.e., $O(\log n)$ bits) message sent to each neighbor

CONGEST model (most standard model of synchronous distributed computation)

Complexity: the <u>number of rounds</u> needed for the computation

Computing eccentricities and the diameter are among the most fundamental tasks

<u>example</u>: at each round, a can send one message to c
b can send one message to c
c can send one message to a, one to b, one to d and one to e
.....



ecc (a) = 3
ecc (b) = 3
ecc (c) = 2
ecc (d) = 3
ecc (e) = 3
ecc (f) = 4
ecc (g) = 4

D = 4

Classical Distributed Computation of the Eccentricity

Let's write n = number of nodes

- ✓ Each node represents a processor (with a unique ID)
- ✓ Each edge represents a classical channel
- ✓ At each round only one short (i.e., $O(\log n)$ bits) message sent to each neighbor

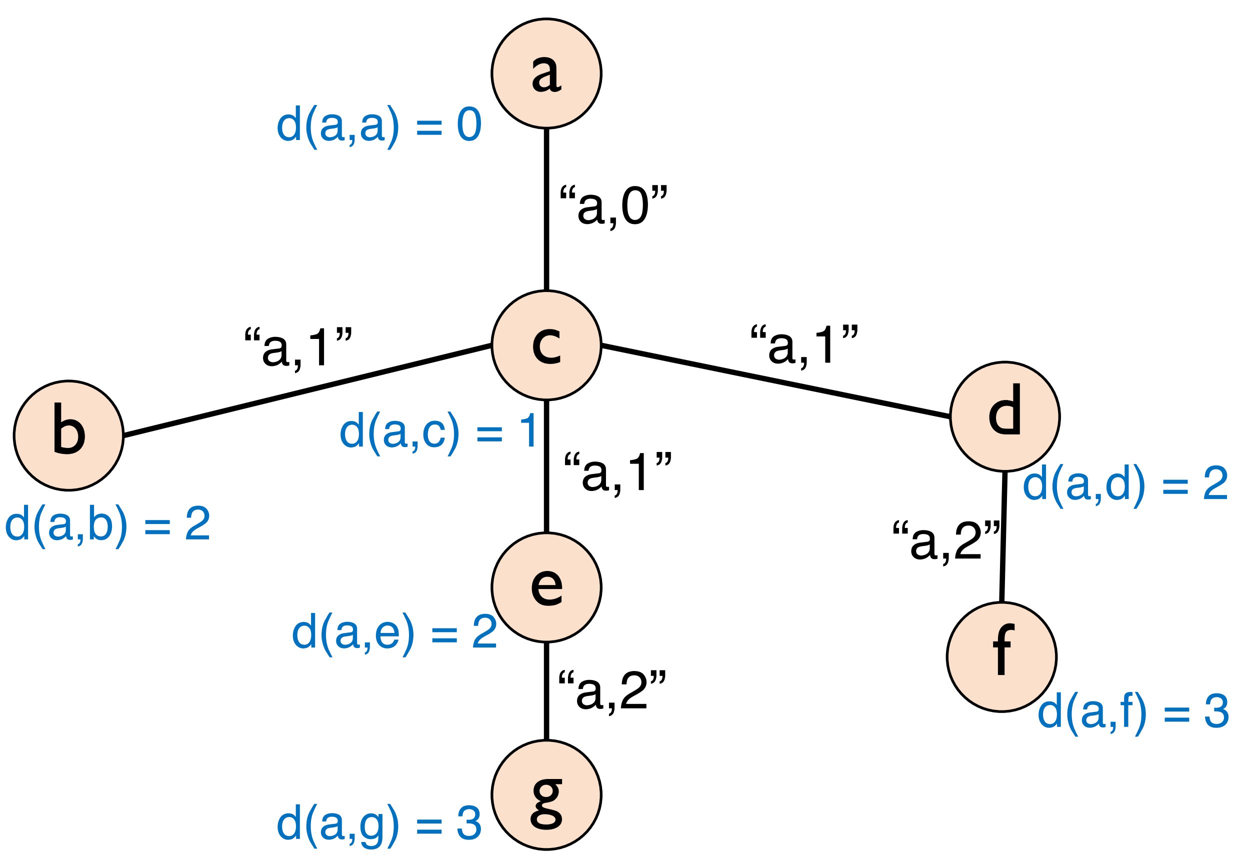
CONGEST model (most standard model of synchronous distributed computation)

Complexity: the <u>number of rounds</u> needed for the computation

Computing eccentricities and the diameter are among the most fundamental tasks

<u>example</u>: computation of ecc (a)

first round of communication
second round of communication
third round of communication



<u>Computation of ecc (u)</u>:

Starting with u, each node broadcasts its distance to u to its neighbors. (Each node knows its distance to u the first time it receives a message.)

complexity: ecc (u) rounds

The nodes then compute the maximum of their distance (easy)

Classical Distributed Computation of the Diameter

Let's write n = number of nodes

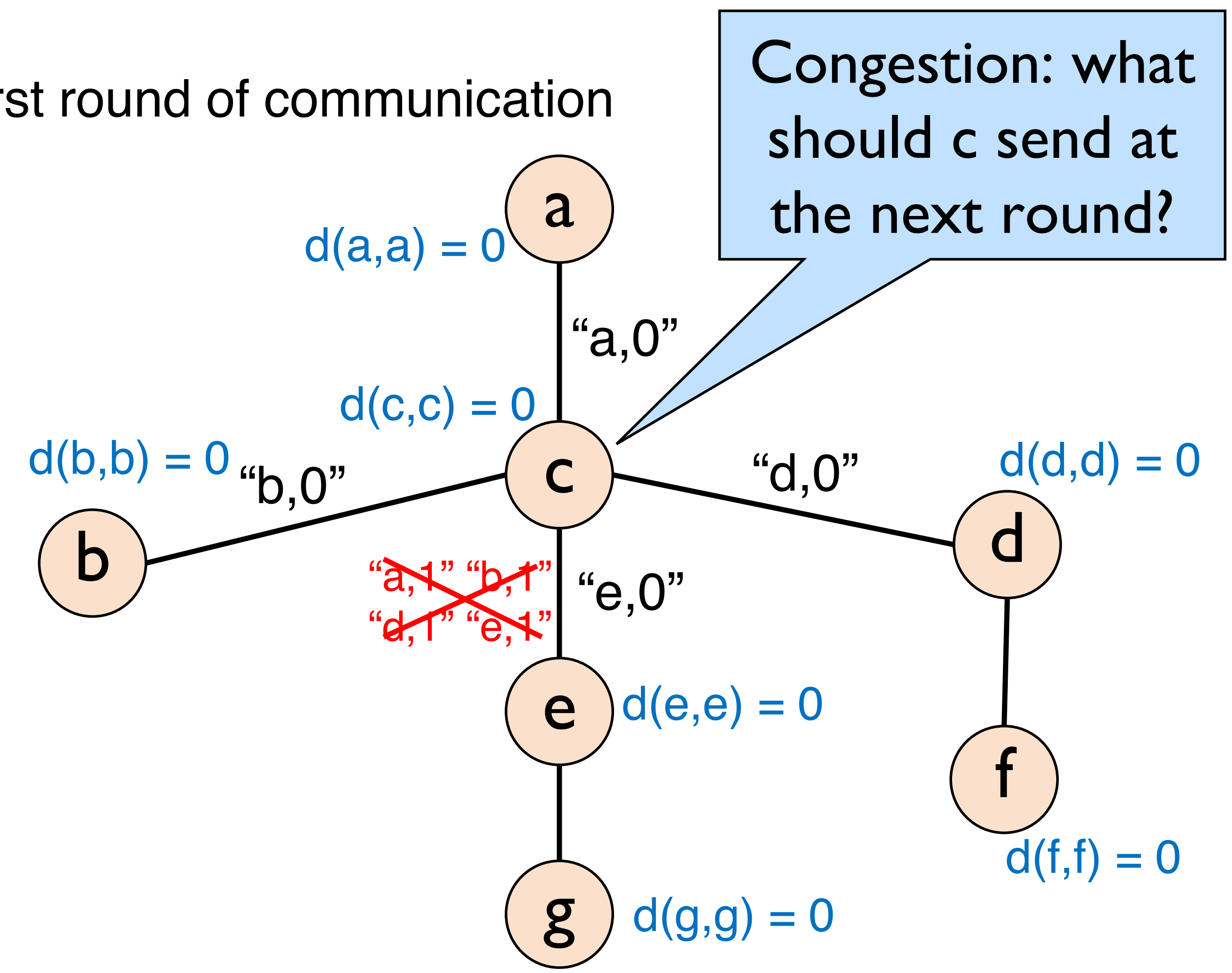
- ✓ Each node represents a processor (with a unique ID)
- ✓ Each edge represents a classical channel
- ✓ At each round only one short (i.e., *O*(log n) bits) message sent to each neighbor

CONGEST model (most standard model of synchronous distributed computation)

Complexity: the <u>number of rounds</u> needed for the computation

Computing eccentricities and the diameter are among the most fundamental tasks

first round of communication



<u>Computation of the diameter D</u>:

All the nodes compute simultaneously their eccentricity <u>using standard techniques to handle congestions</u>

complexity: Θ(n) rounds (even if D is constant)

[Holzer+12, Peleg+12]

Output the maximum eccentricity

Computation of the Diameter

main result: sublinear-round quantum computation of the diameter whenever D=o(n)
(our algorithm uses $O((\log n)^2)$ qubits of quantum memory per node)

first gap between classical and quantum for an important problem in the CONGEST model of distributed computing

	Classical	Quantum (our results)
Exact computation (upper bounds)	$O(n)$ [Holzer+12, Peleg+12]	$O(\sqrt{nD})$
Exact computation (lower bounds)	$\widetilde{\Omega}(n)$ [Frischknecht+12]	$\widetilde{\Omega}(\sqrt{n}+D)$ [unconditional] $\widetilde{\Omega}(\sqrt{nD})$ [conditional]

number of rounds needed to compute the diameter (n: number of nodes, D: diameter)

number of rounds needed to compute the diameter (n: number of nodes, D: diameter)

the tilde notation removes polylog(n) factors

condition: holds for algorithms using only polylog(n) qubits of memory per node

3/2-approximation (upper bounds)	$O(\sqrt{n}+D)$ [Lenzen+13,Holzer+14]	$O(\sqrt[3]{nD}+D)$
(3/2-ε)-approximation (lower bounds)	$\widetilde{\Omega}(n)$ [Holzer+12,Abboud+16]	$\widetilde{\Omega}(\sqrt{n}+D)$ [unconditional]

Our Upper Bound

main result: sublinear-round quantum computation of the diameter whenever D=o(n)

(our algorithm uses $O((\log n)^2)$ qubits of quantum memory per node)

first gap between classical and quantum for an important problem in the CONGEST model of distributed computing

	Classical	Quantum (our results)
Exact computation (upper bounds)	$O(n)$ [Holzer+12, Peleg+12]	$\boxed{O(\sqrt{nD})}$

number of rounds needed to compute the diameter (n: number of nodes, D: diameter)

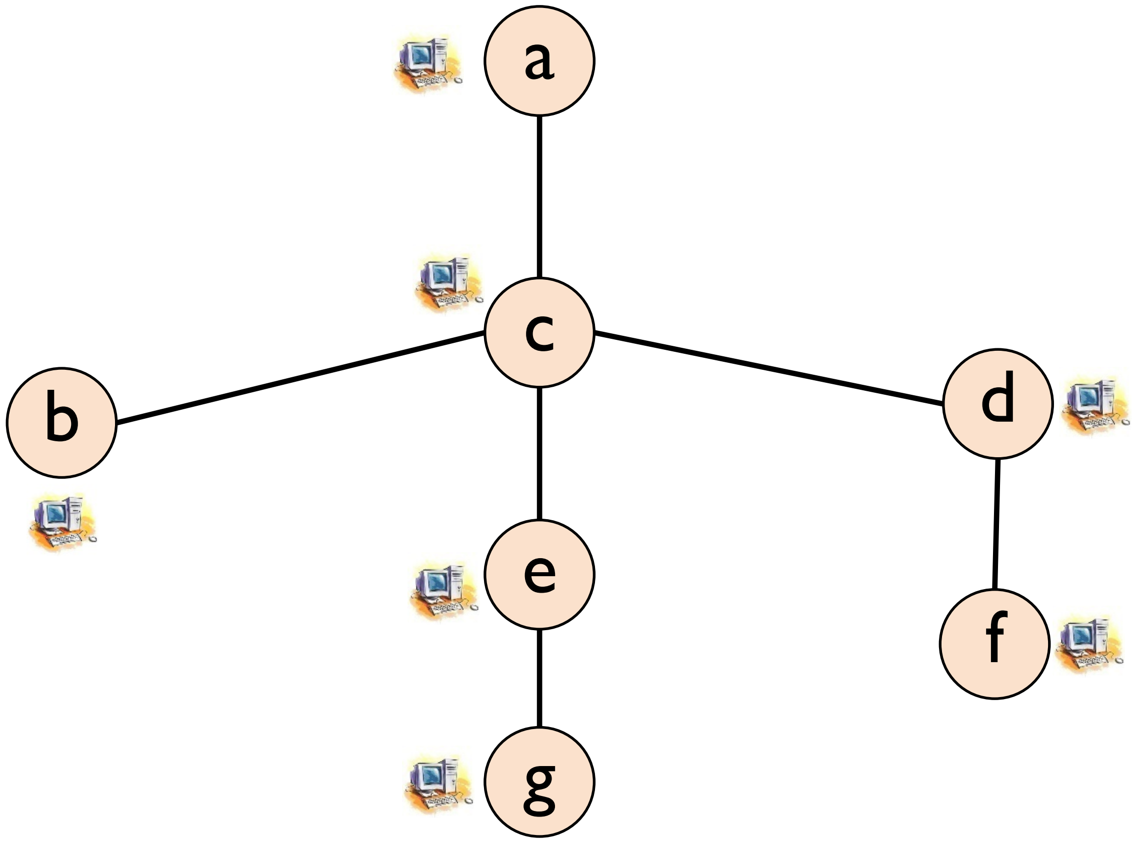
Quantum Distributed Computation of the Diameter

Let's write n = number of nodes

- ✓ Each node represents a quantum processor (with a unique ID)
- ✓ Each edge represents a quantum channel
- ✓ At each round only one short ($O(\log n)$ qubits) message sent to each neighbor

quantum CONGEST model

Complexity: the <u>number of rounds</u> needed for the computation



Quantum Distributed Computation of the Diameter

Let's write n = number of nodes

- ✓ Each node represents a quantum processor (with a unique ID)
- ✓ Each edge represents a quantum channel
- ✓ At each round only one short ($O(\log n)$ qubits) message sent to each neighbor

quantum CONGEST model

Complexity: the number of rounds needed for the computation

Computation of the diameter (decision version)

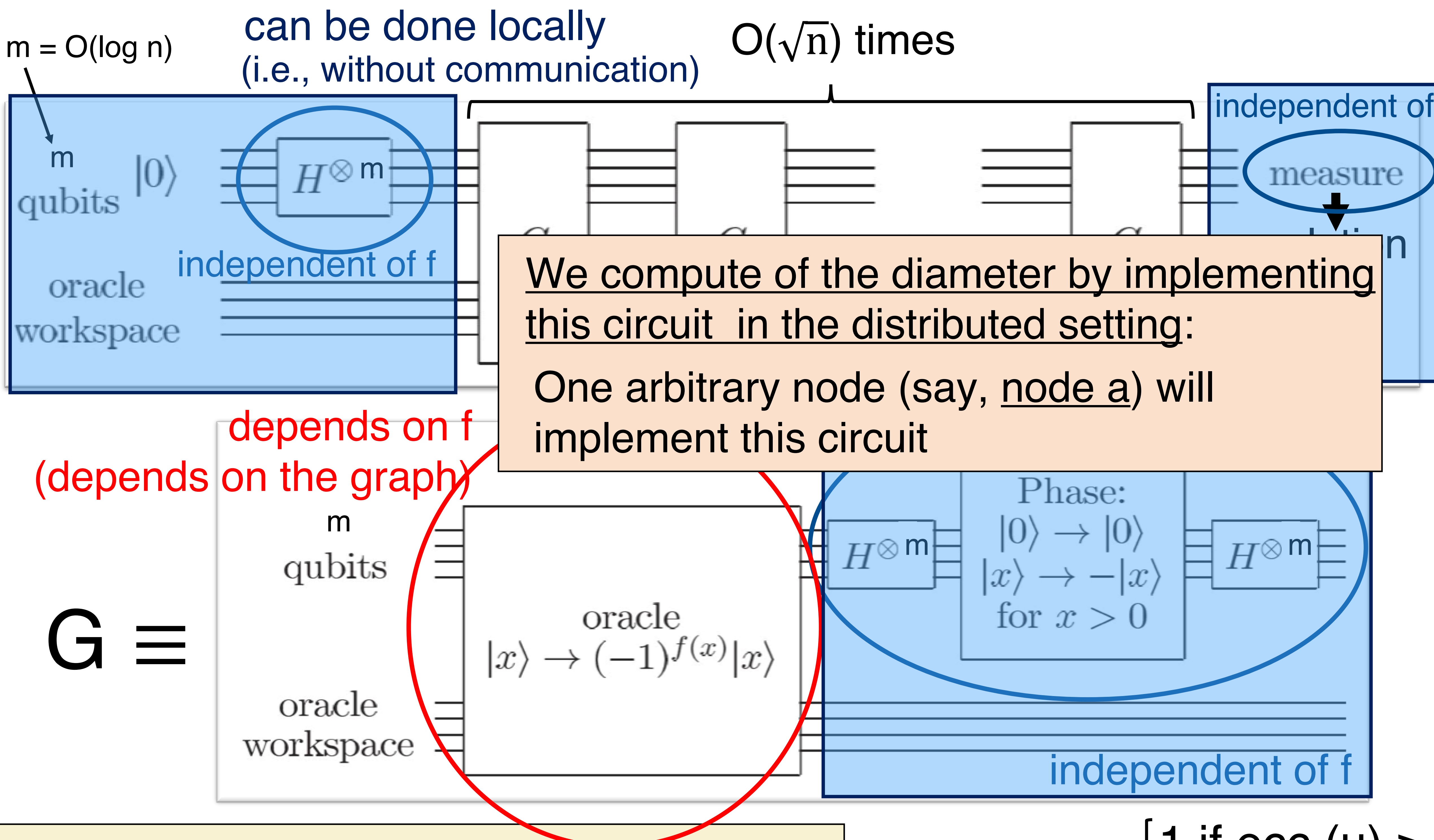
Given an integer d, decide if $\underbrace{\text{diameter} \geq d}_{\text{there is a vertex u such that ecc (u)} \geq d}$

This is a search problem, so we can try to use Grover search:

Find an element $u \in V$ such that $f(u) = 1$

$$\text{with } f(u) = \begin{cases} 1 \text{ if ecc }(u) \geq d \\ 0 \text{ otherwise} \end{cases}$$

Usual Grover Algorithm (from Nielsen-Chuang, page 251)



We compute of the diameter by implementing this circuit in the distributed setting:

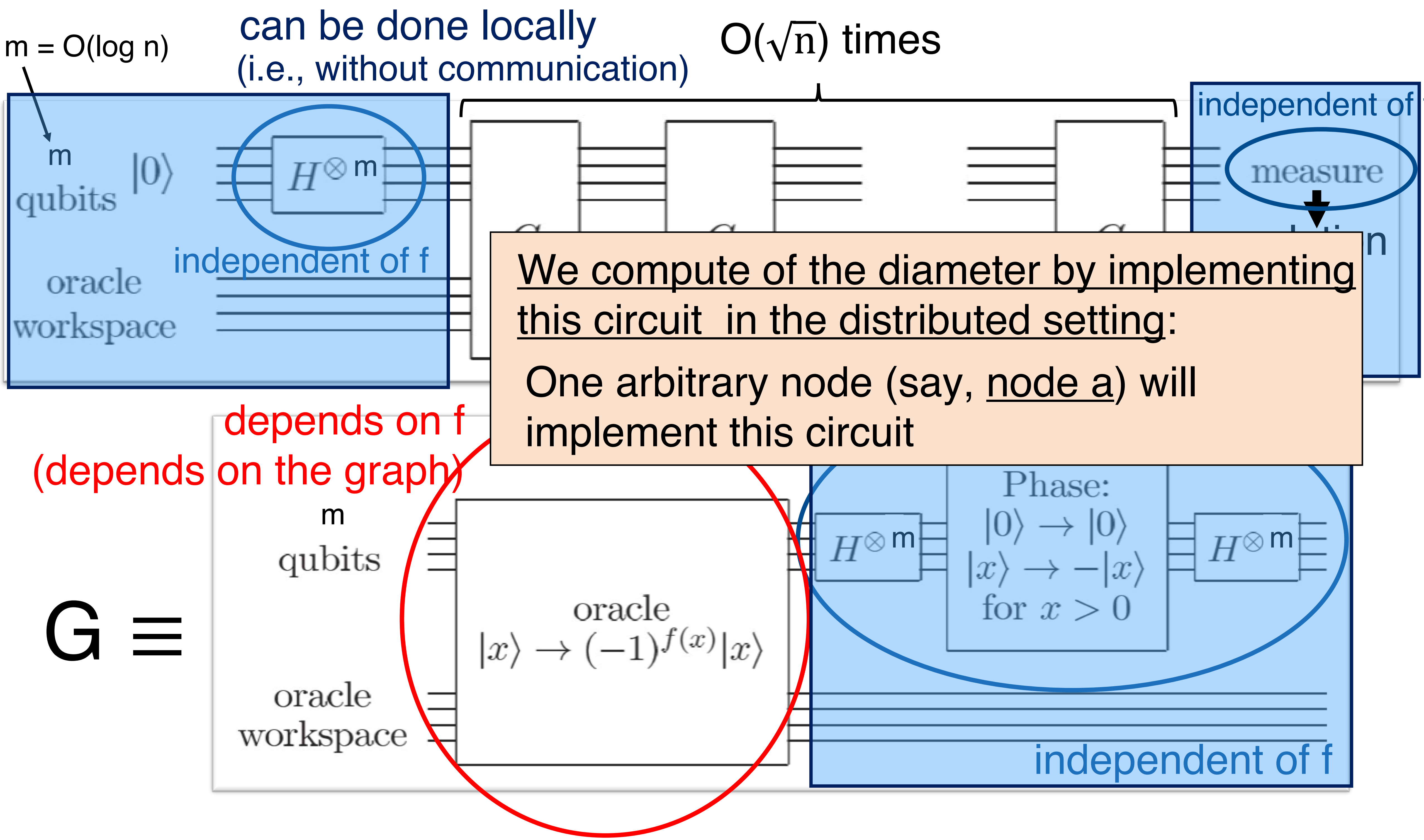
One arbitrary node (say, node a) will implement this circuit

Find an element $u \in V$ such that f(u) = 1

with $f(u) = \begin{cases} 1 \text{ if ecc } (u) \geq d \\ 0 \text{ otherwise} \end{cases}$

remember: n = |V|

Usual Grover Algorithm (from Nielsen-Chuang, page 251)



To implement the oracle, the node a needs to communicate with the other nodes

Total number of rounds of communication = O($\sqrt{n}$ x number of rounds to implement the oracle)

Implementati

Node a introduces 1 register $\sum_{u\in V}\alpha_u|u\rangle_{\mathrm{a}}|0\rangle$

Node a applies CNOTS $\sum_{u\in V}\alpha_u|u\rangle_{\mathrm{a}}|u\rangle$

Node a sends the second register to c $\sum_{u\in V}\alpha_u|u\rangle_{\mathrm{a}}|u\rangle_{\mathrm{c}}$

Node c introduces 3 registers $\sum_{u\in V}\alpha_u|u\rangle_{\mathrm{a}}\,|u\rangle_{\mathrm{c}}|0\rangle\ \ |0\rangle\ \ |0\rangle$

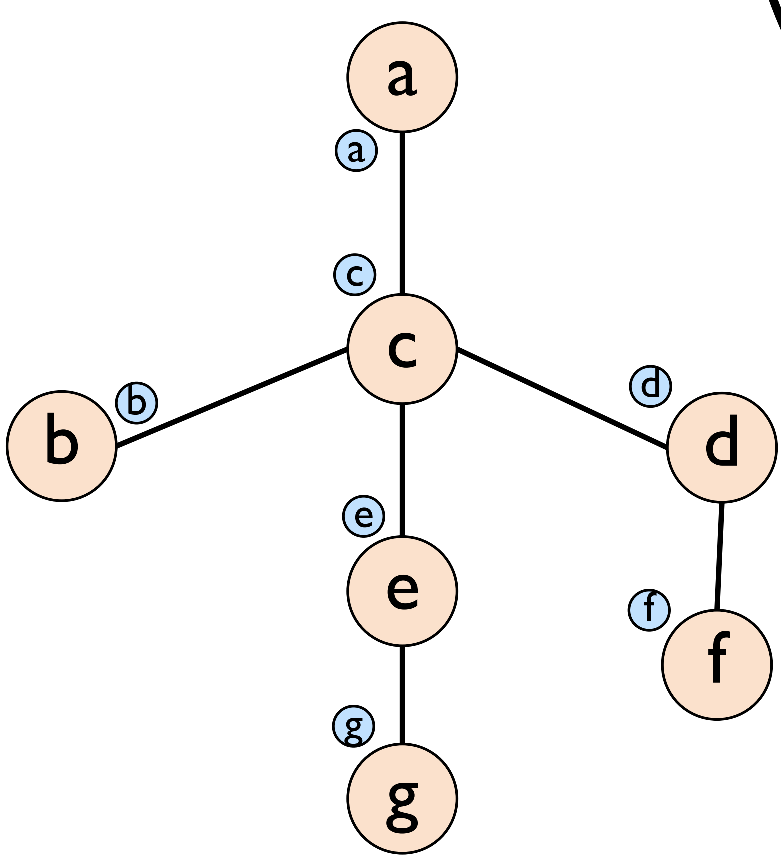
Node c applies CNOTS $\sum_{u\in V}\alpha_u|u\rangle_{\mathrm{a}}\,|u\rangle_{\mathrm{c}}|u\rangle\ \ |u\rangle\ \ |u\rangle$

Node c sends the registers to b,e,d $\sum_{u\in V}\alpha_u|u\rangle_{\mathrm{a}}\,|u\rangle_{\mathrm{c}}|u\rangle_{\mathrm{b}}|u\rangle_{\mathrm{e}}|u\rangle_{\mathrm{d}}$

……

$$\sum_{u\in V}\alpha_u|u\rangle|0\rangle$$

V={a,b,c,d,e,f,g}



Initially node a owns $\sum_{u\in V}\alpha_u|u\rangle_{\mathrm{a}}$

1. “Broadcast” this state, which gives [ecc(a) ≤ D rounds]

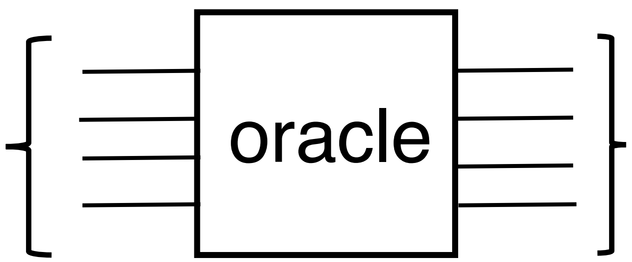
$$\sum_{u\in V}\alpha_u|u\rangle_{\mathrm{a}}|u\rangle_{\mathrm{b}}|u\rangle_{\mathrm{c}}|u\rangle_{\mathrm{d}}|u\rangle_{\mathrm{e}}|u\rangle_{\mathrm{f}}|u\rangle_{\mathrm{g}}$$

2. The nodes implements the classical protocol for computing the eccentricity, which gives [≤ D rounds]

$$\sum_{u\in V}\alpha_u|u\rangle_{\mathrm{a}}|u\rangle_{\mathrm{b}}|u\rangle_{\mathrm{c}}|u\rangle_{\mathrm{d}}|u\rangle_{\mathrm{e}}|u\rangle_{\mathrm{f}}\,|u\rangle_{\mathrm{g}}|ecc(u)\rangle_{\mathrm{a}}$$

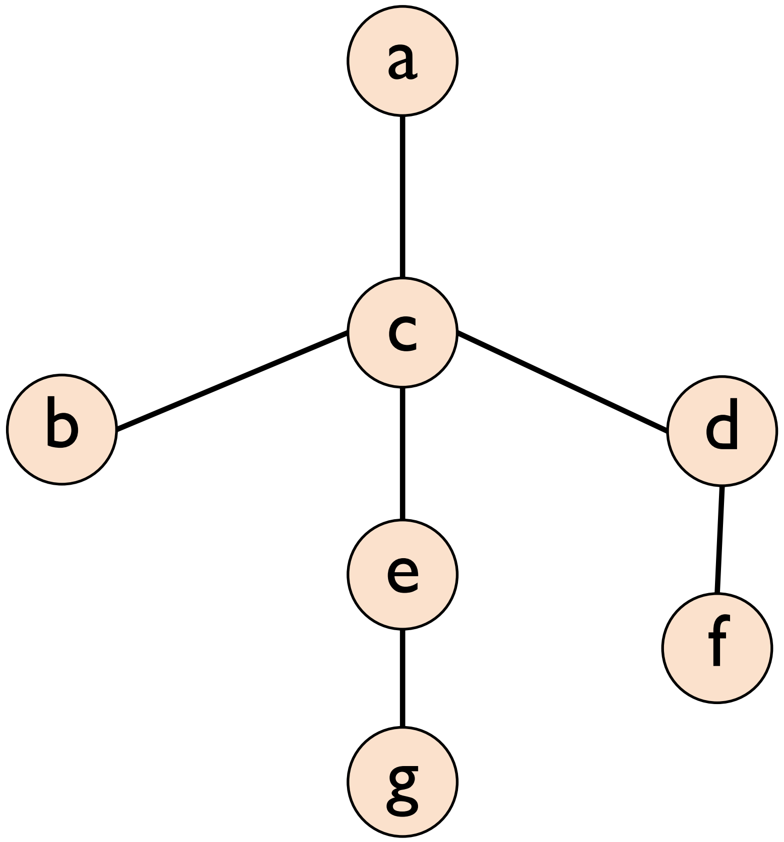
Implementation of the Oracle in O(D) rounds

$$\sum_{u\in V} \alpha_u |u\rangle_{\mathrm{a}} |0\rangle_{\mathrm{a}}$$



$$\sum_{u\in V} \alpha_u |u\rangle_{\mathrm{a}} |ecc(u)\rangle_{\mathrm{a}}$$

V={a,b,c,d,e,f,g}



Initially node a owns $\sum_{u\in V} \alpha_u |u\rangle_{\mathrm{a}}$

1. "Broadcast" this state, which gives [ecc(a) ≤ D rounds]

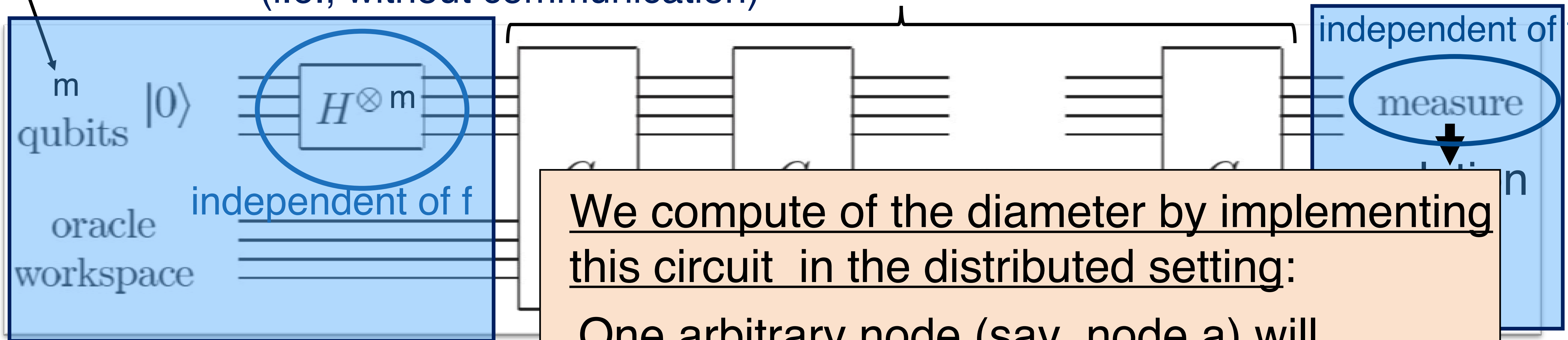
$$\sum_{u\in V} \alpha_u |u\rangle_{\mathrm{a}} |u\rangle_{\mathrm{b}} |u\rangle_{\mathrm{c}} |u\rangle_{\mathrm{d}} |u\rangle_{\mathrm{e}} |u\rangle_{\mathrm{f}} |u\rangle_{\mathrm{g}}$$

2. The nodes implements the classical protocol for computing the eccentricity, which gives [≤ D rounds]

$$\sum_{u\in V} \alpha_u |u\rangle_{\mathrm{a}} |u\rangle_{\mathrm{b}} |u\rangle_{\mathrm{c}} |u\rangle_{\mathrm{d}} |u\rangle_{\mathrm{e}} |u\rangle_{\mathrm{f}} |u\rangle_{\mathrm{g}} |ecc(u)\rangle_{\mathrm{a}}$$

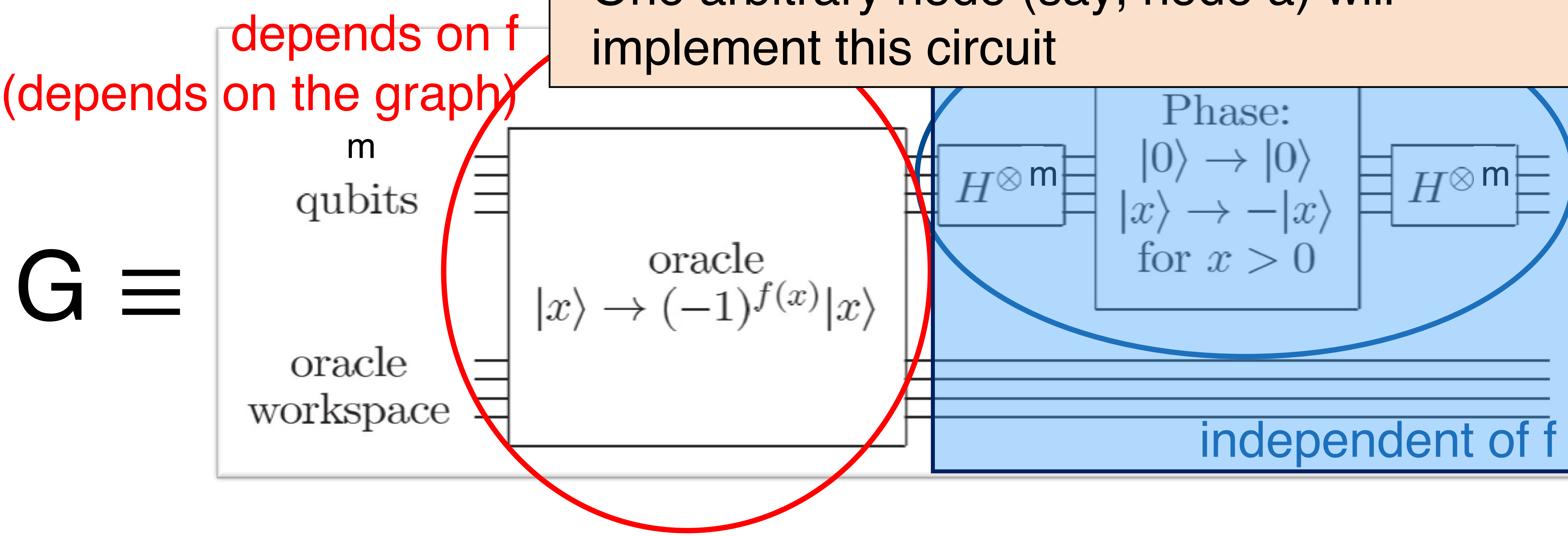
3. The nodes revert Step 1 [ecc(a) ≤ D rounds]

Usual Grover Algorithm (from Nielsen-Chuang, page 251)



We compute of the diameter by implementing this circuit in the distributed setting:

One arbitrary node (say, node a) will implement this circuit



To implement the oracle, the node a needs to communicate with the other nodes

Total number of rounds of communication = O($\sqrt{n}$ x number of rounds to implement the oracle)
= O($\sqrt{n}$ x D)

The Upper Bound

- ✓ We have just described a O($\sqrt{n}$ x D)-round quantum distributed algorithm for computing (with high probability) the diameter
- ✓ With further work, the complexity can be reduced to O($\sqrt{nD}$) rounds

	Classical	Quantum (our results)
Exact computation (upper bounds)	$O(n)$ [Holzer+12, Peleg+12]	$O(\sqrt{nD})$

The Lower Bounds

	Classical	Quantum (our results)
Exact computation (lower bounds)	$\widetilde{\Omega}(n)$ [Frischknecht+12]	$\widetilde{\Omega}(\sqrt{n}+D)$ [unconditional] $\widetilde{\Omega}(\sqrt{nD})$ [conditional]

via two-party communication complexity

<u>classical lower bound</u>

- ✓ reduce DISJOINTNESS to the distributed computation of diameter [Frischknecht+12]
- ✓ the (two-party) communication complexity of DISJOINTNESS is Ω(n) bits [Kalyanasundaram+92]

<u>unconditional quantum lower bound</u>

- ✓ same reduction from DISJOINTNESS to the diameter
- ✓ the (two-party) communication complexity of DISJOINTNESS is $\Omega(\sqrt{n})$ qubits [Razborov03]

<u>conditional quantum lower bound</u>

- ✓ if the quantum distributed algorithm for diameter <u>uses few quantum memory</u> per node, then the computation of DISJOINTNESS can be done using few messages
- ✓ the (two-party) r-message communication complexity of DISJOINTNESS is $\Omega(r+n/r)$ qubits [Braverman+15]

Quantum Computation of the Diameter: Summary

main result: sublinear-round quantum computation of the diameter

first gap between classical and quantum for an important problem in the CONGEST model of distributed computing

	Classical	Quantum (our results)
Exact computation (upper bounds)	$O(n)$ [Holzer+12, Peleg+12]	$O(\sqrt{nD})$
Exact computation (lower bounds)	$\widetilde{\Omega}(n)$ [Frischknecht+12]	$\widetilde{\Omega}(\sqrt{n}+D)$ [unconditional] $\widetilde{\Omega}(\sqrt{nD})$ [conditional]

number of rounds needed to compute the diameter (n: number of nodes, D: diameter)

3/2-approximation (upper bounds)	$O(\sqrt{n}+D)$ [Lenzen+13,Holzer+14]	$O(\sqrt[3]{nD}+D)$
(3/2-ε)-approximation (lower bounds)	$\widetilde{\Omega}(n)$ [Holzer+12,Abboud+16]	$\widetilde{\Omega}(\sqrt{n}+D)$ [unconditional]

- ✓ Our upper bounds are obtained by showing how to implement quantum search in a distributed setting
- ✓ Interesting research direction: find other applications of this technique

量子アルゴリズムの理論研究の潮流

引き続き、大規模量子コンピュータ向けの量子アルゴリズムの開発

量子分散計算の枠組みで研究

F. Le Gall and F. Magniez. **Sublinear-Time Quantum Computation of the Diameter in Distributed Networks**. Proceedings of the 37th ACM Symposium on Principles of Distributed Computing (PODC 2018).
Also arXiv: 1804.02917.

分散計算の重要な問題に対して、
古典分散アルゴリズムより高速な量子分散アルゴリズムを初めて与えた

量子アルゴリズムの理論研究の潮流

引き続き、大規模量子コンピュータ向けの量子アルゴリズムの開発

量子分散計算の枠組みで研究

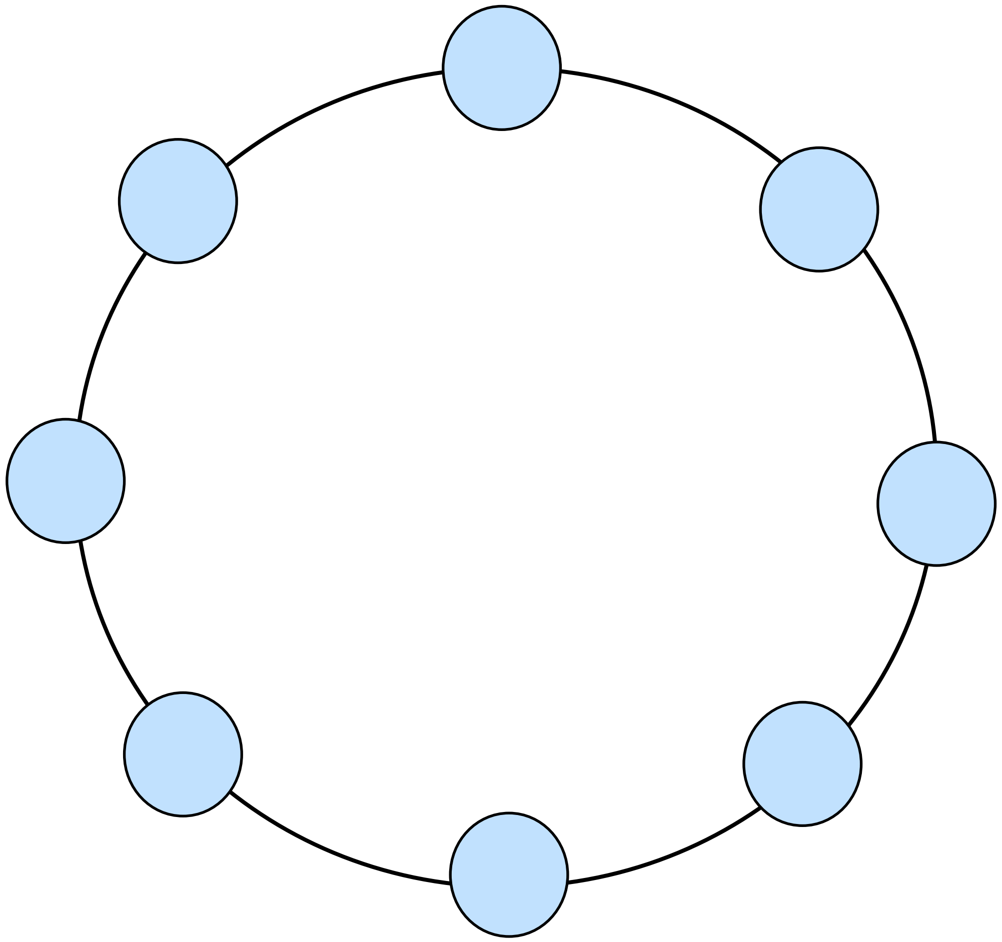
QUANTUM SUPREMACY
（小〜中規模量子コンピュータの優位性の確立）

量子分散計算を用いて研究

S. Bravyi, D. Gosset and R. König. **Quantum Advantage with Shallow Circuits.** ArXiv: 1704.00690. Plenary talk at QIP 2018.

リング状のグラフ状態

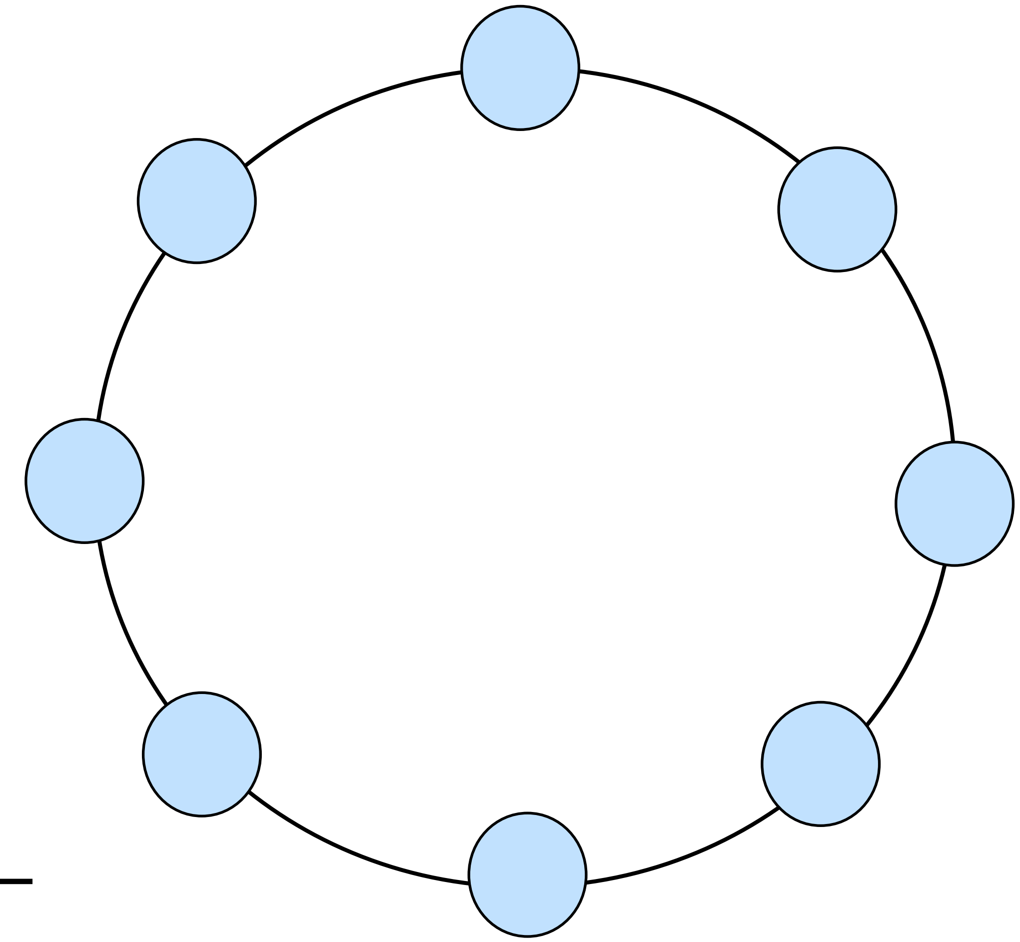
n量子ビットのリング状のグラフ状態を考えよう

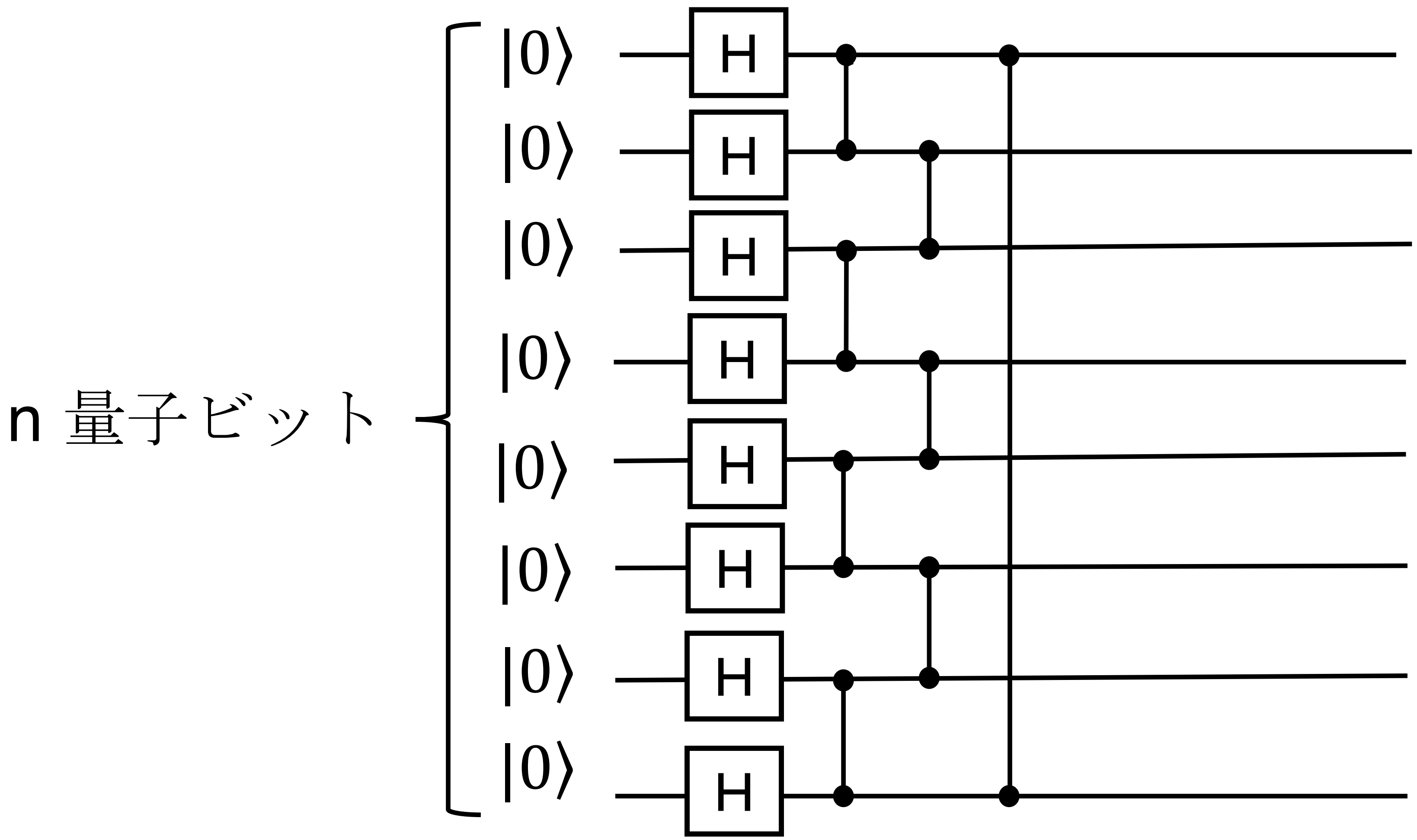


リング状のグラフ状態

n量子ビットのリング状のグラフ状態を考えよう

各量子ビットを（あるルールに従って）X基底かY基底で観測しよう

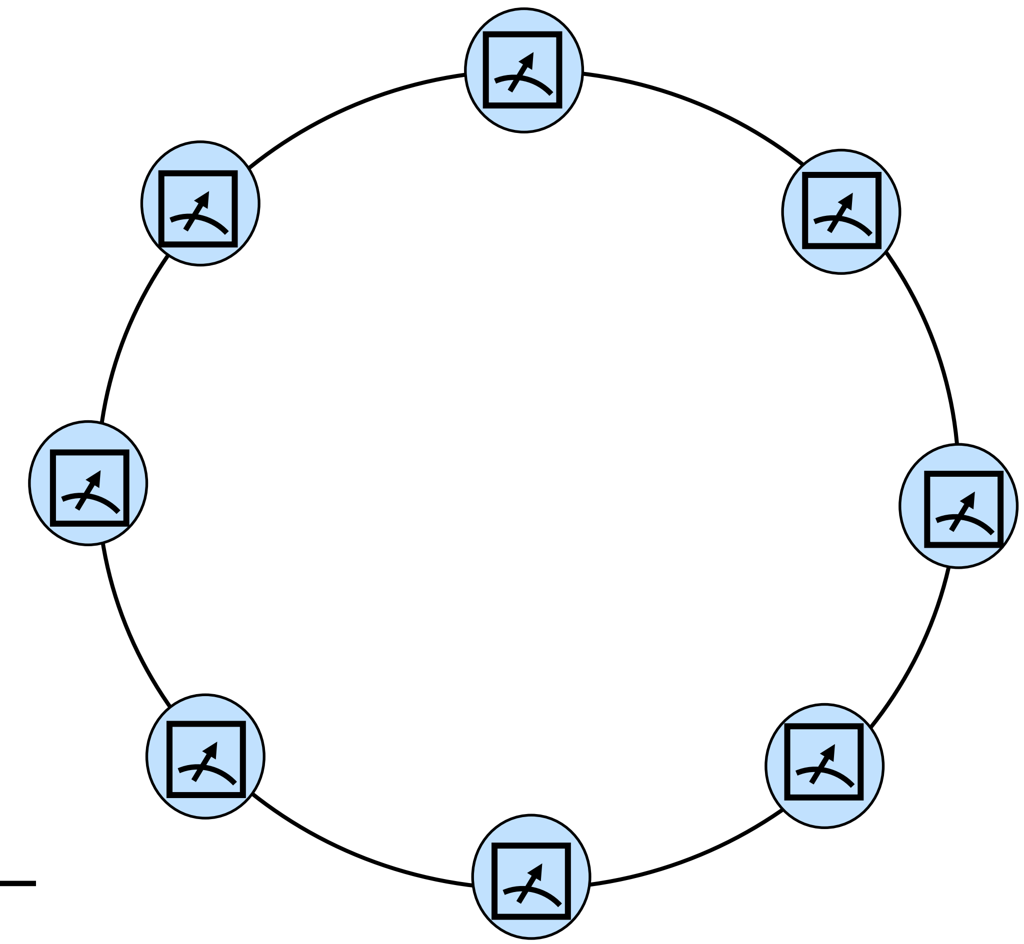




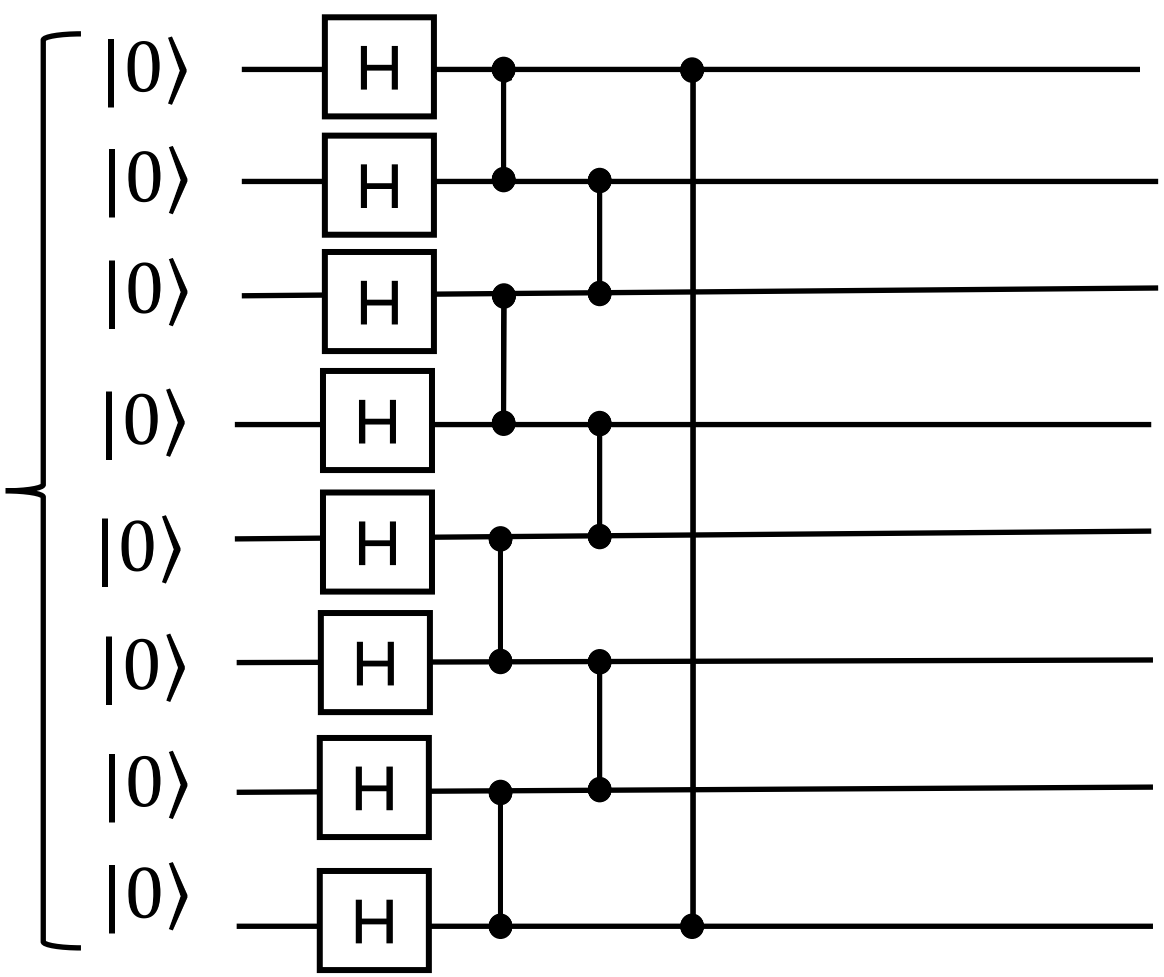
リング状のグラフ状態

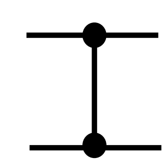
n量子ビットのリング状のグラフ状態を考えよう

各量子ビットを（あるルールに従って）X基底かY基底で観測しよう



n 量子ビット



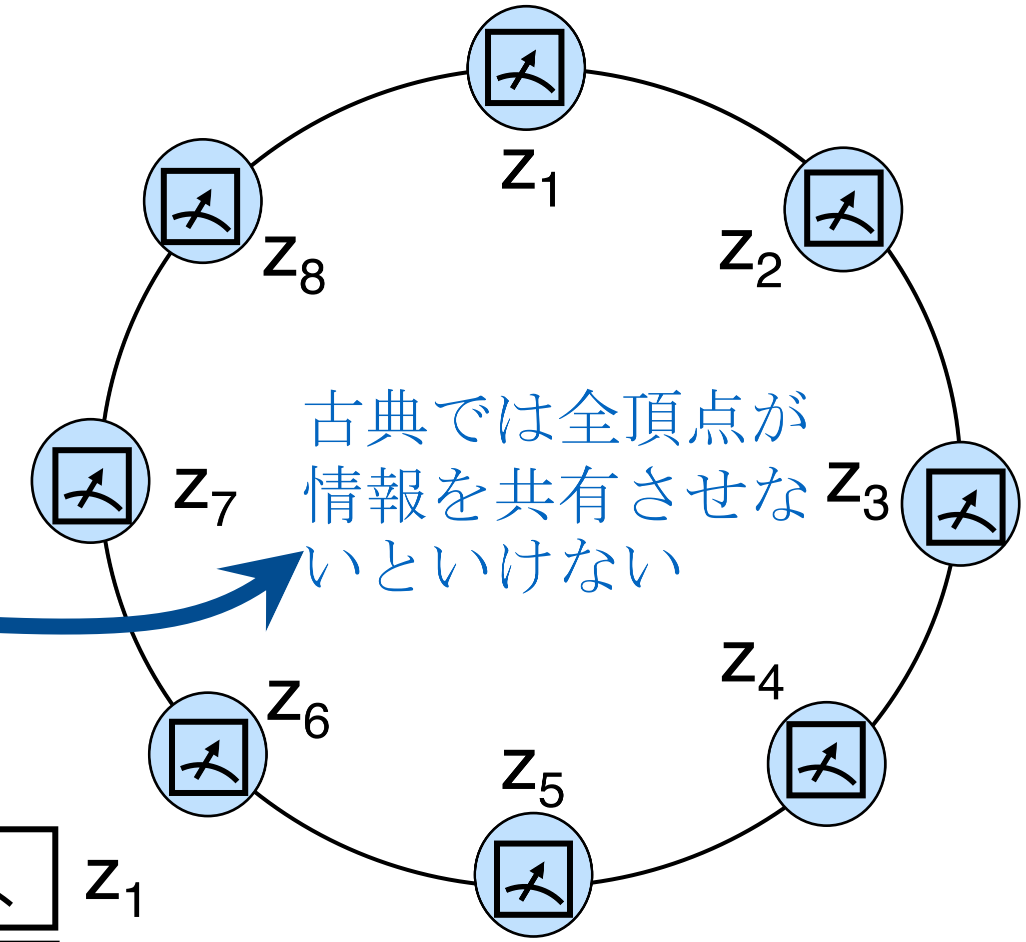


≡ Controlled Z gate

リング状のグラフ状態

n量子ビットのリング状のグラフ状態を考えよう

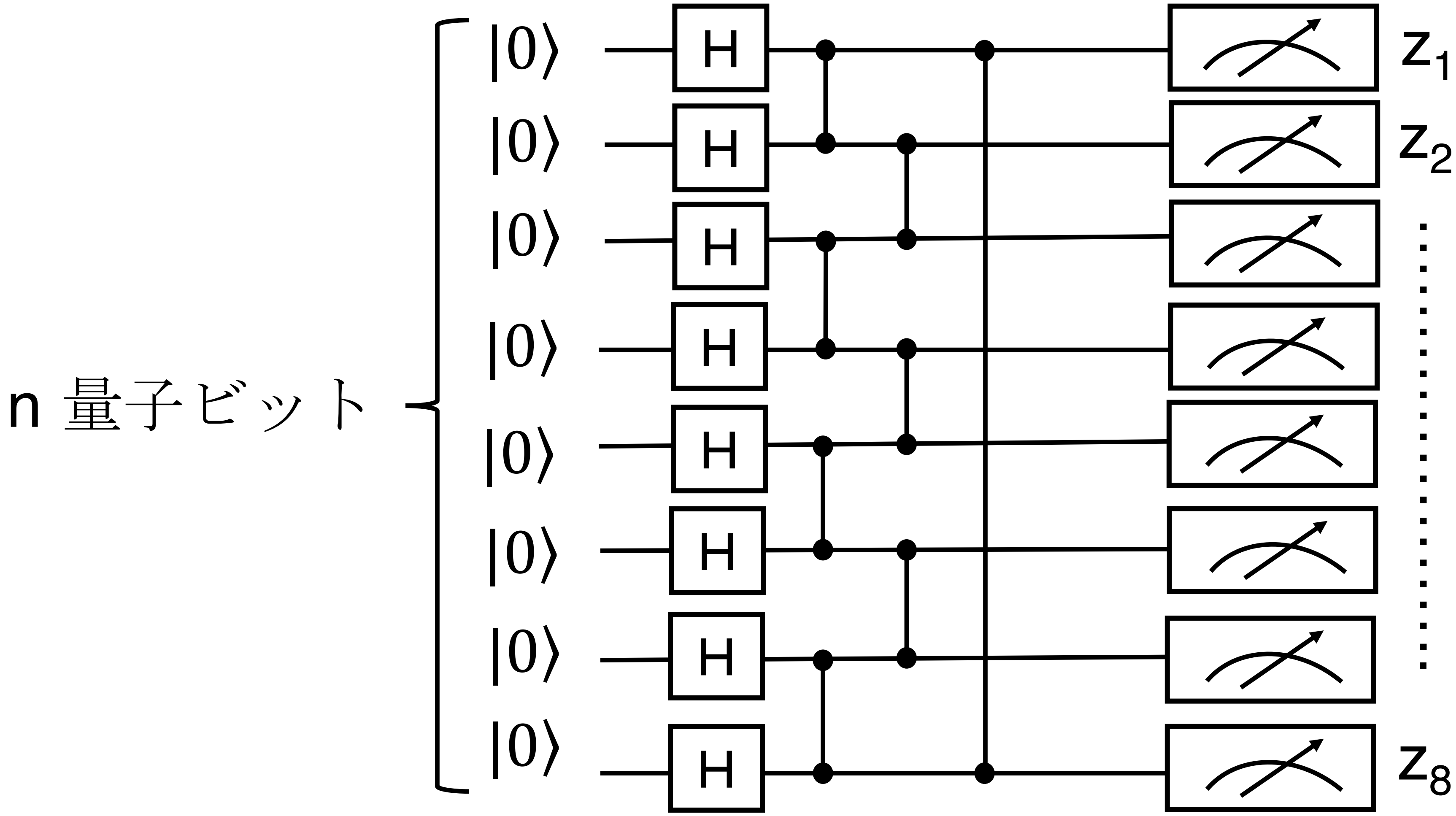
各量子ビットを（あるルールに従って）X基底かY基底で観測しよう



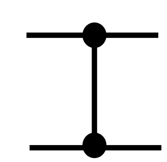
Theorem [Barrett et al. 06]

> 観測結果の分布を古典で生成するために、$\Omega(n)$ラウンドの通信が必要

[Le Gall, Nishimura, Rosmanis 18]：観測結果の分布を古典で近似するために、$\Omega(n)$ラウンドの通信が必要



量子では通信はもちろん必要ない！



≡ Controlled Z gate

[Bravyi et al. 17]：定数深さ量子回路の優位性の無条件証明

（今までの証明は P≠NP のような条件を仮定）

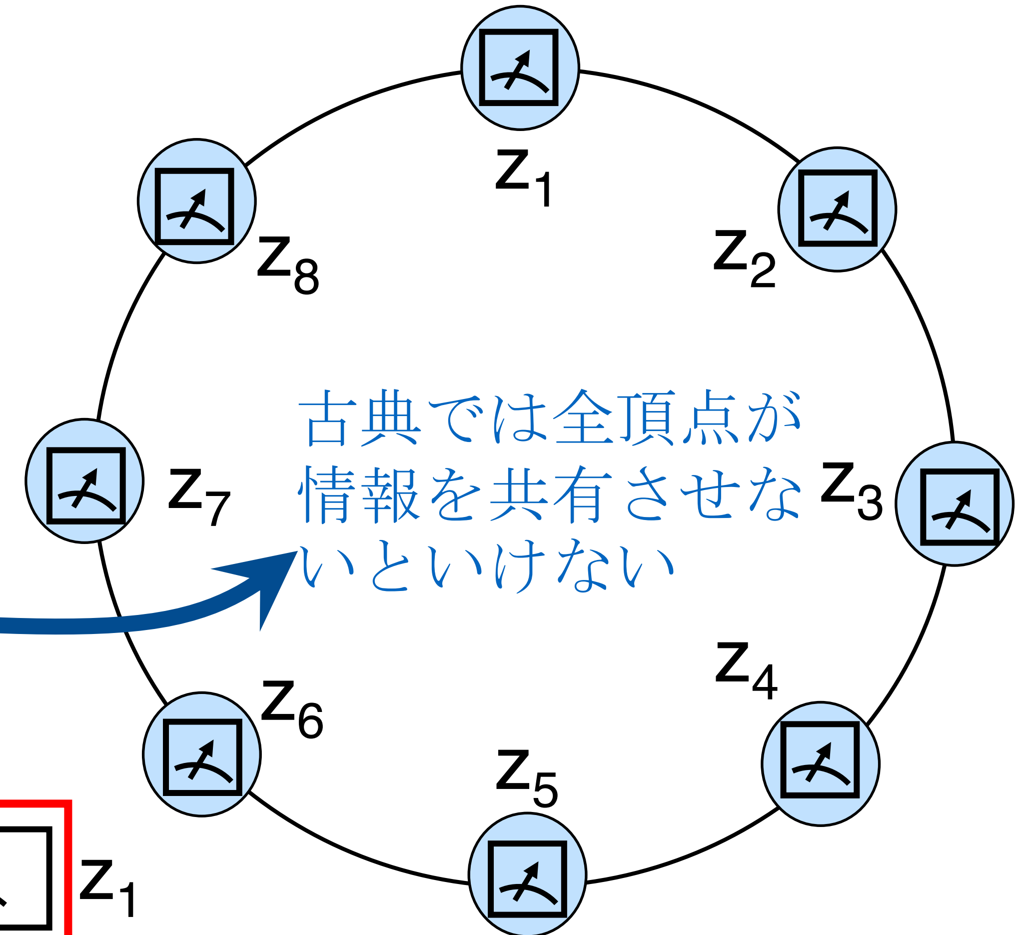
n量子ビットのリング状のグラフ状態を考えよう

各量子ビットを（あるルールに従って） X基底かY基底で観測しよう

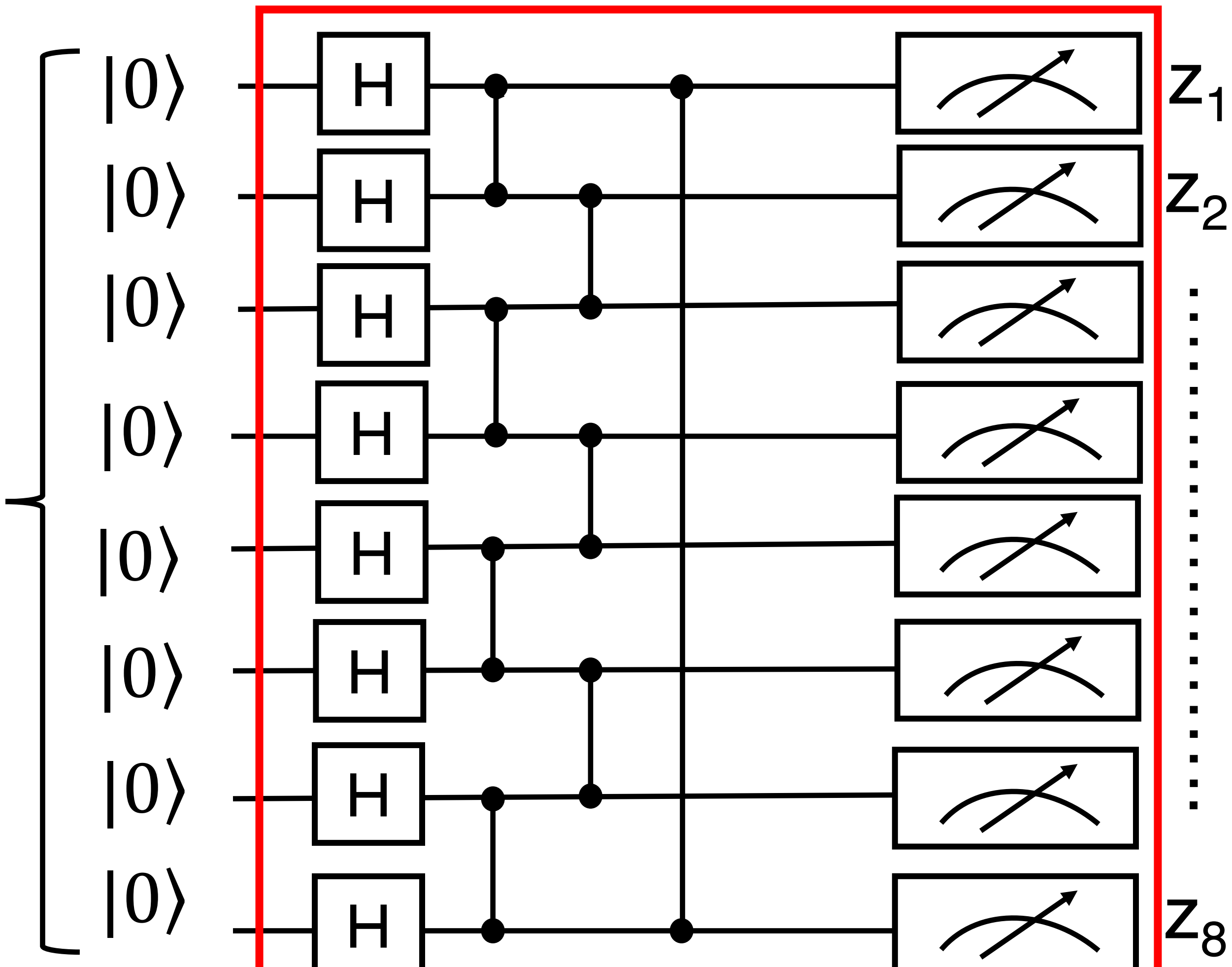
同じ分布を生成するため、
古典では深さΩ(log n)の回路が必要
（全プレイヤーが情報を共有するために深さΩ(log n)が必要）

Theorem [Barrett et al. 06]

観測結果の分布を古典で生成するために、
Ω(n)ラウンドの通信が必要



n 量子ビット



定数深さの量子回路
で生成

まとめ：量子計算と分散計算

引き続き、大規模量子コンピュータ向けの量子アルゴリズムの開発

量子分散計算の枠組みで研究

分散計算の重要な問題に対して、
古典分散アルゴリズムより高速な量子分散アルゴリズムを初めて与えた

[Le Gall and Magniez 2018]

QUANTUM SUPREMACY
（小〜中規模量子コンピュータの優位性の確立）

量子分散計算を用いて研究

定数深さ量子回路の優位性の無条件証明

[Bravyi, Gosset and König 2017]

未解決：より計算能力の高い量子回路のクラスの優位性の無条件証明