Simplest quantum algorithms

Julius Ruseckas

March 4, 2020

Baltic Institute of Advanced Technology

Outline

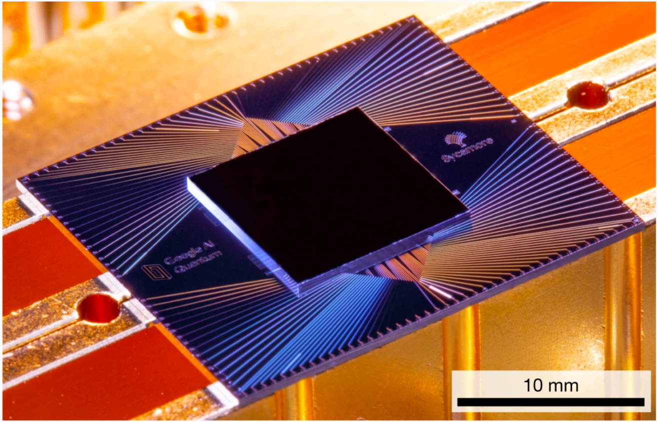
Introduction

Quote

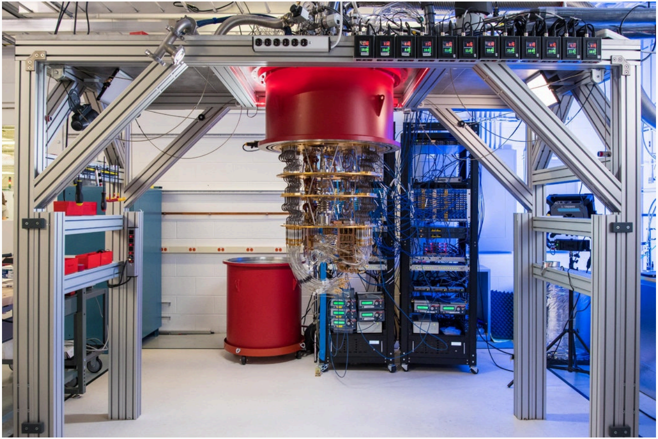
…quantum mechanics becomes elegant and intelligible only after attempts to describe it in words are abandoned

Freeman Dyson

Quantum computer



Google's Sycamore chip...



...and supporting infrastructure

Classical vs Quantum

- Operations in quantum computer are reversible
- Every irreversible classical computation can be made reversible
- Quantum computer can execute all classical algorithms
- Quantum algorithms: use some essential feature of quantum computation

Quantum algorithms

- Algorithms based on the quantum Fourier transform
  - Deutsch–Jozsa algorithm
  - Bernstein–Vazirani algorithm
  - Simon's algorithm
  - Quantum phase estimation algorithm
  - Shor's algorithm
  - Hidden subgroup problem
  - Boson sampling problem
  - Estimating Gauss sums
  - Fourier fishing and Fourier checking
- Algorithms based on amplitude amplification
  - Grover's algorithm
  - Quantum counting

- Algorithms based on quantum walks
  - Element distinctness problem
  - Triangle-finding problem
  - Formula evaluation
  - Group commutativity
- BQP-complete problems
  - Computing knot invariants
  - Quantum simulation
  - Solving a linear systems of equations

Hadamard transform:

$$\boldsymbol{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Action:

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

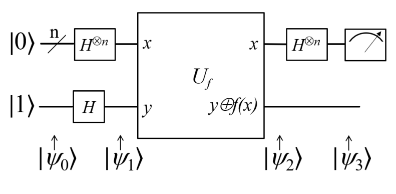
Deutsch's algorithm

Proposed in 1985

Problem

We have a function f taking one bit and returning one bit.

We need to know if $f(0) = f(1)$.



Algorithm

1. We are given a quantum implementation $\boldsymbol{U}_f$ of the function f that maps $|x\rangle|y\rangle$ to $|x\rangle|f(x) \oplus y\rangle$:

$$\boldsymbol{U}_f|x\rangle|y\rangle = |x\rangle|f(x) \oplus y\rangle$$

   The second qubit is flipped if f acting on the first qubit is 1, and remains the same if f acting on the first qubit is 0

2. We begin with the two-qubit state $|0\rangle|1\rangle$ and apply a Hadamard transform to each qubit:

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

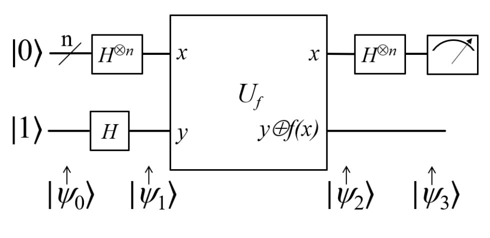
We have:

$$\boldsymbol{U}_f|x\rangle\frac{1}{\sqrt{2}}\big(|0\rangle-|1\rangle\big)=|x\rangle\frac{1}{\sqrt{2}}\big(|f(x)\rangle-|1\oplus f(x)\rangle\big)$$
$$=|x\rangle(-1)^{f(x)}\frac{1}{\sqrt{2}}\big(|0\rangle-|1\rangle\big)$$

Thus

$$\boldsymbol{U}_f\frac{1}{2}\big(|0\rangle+|1\rangle\big)\big(|0\rangle-|1\rangle\big)$$
$$=\frac{1}{2}\big[(-1)^{f(0)}|0\rangle+(-1)^{f(1)}|1\rangle\big]\big(|0\rangle-|1\rangle\big)$$
$$=(-1)^{f(0)}\frac{1}{2}\big[|0\rangle+(-1)^{f(0)\oplus f(1)}|1\rangle\big]\big(|0\rangle-|1\rangle\big)$$

3. We apply Hadamard transform to the first qubit.
4. Measurement of the first qubit. The result is 0 if and only if $f(0) = f(1)$



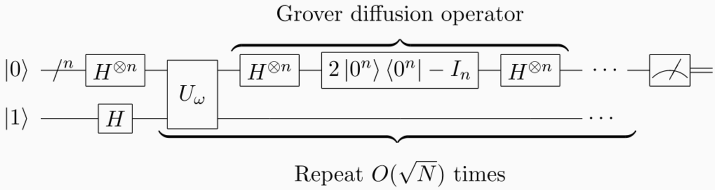
Grover's algorithm

Proposed in 1996

Problem

We need to find the unique input to a black box function that produces a particular output value.

Inverting a function is related to the searching of a database



Algorithm

1. We have a function f_ω where $f_\omega(x) = 1$ if x satisfies the search criterion ω:

$$f_\omega(x) = \begin{cases} 1\,, & x = \omega\,, \\ 0\,, & x \neq \omega\,. \end{cases}$$

   x can aquire N values.

   We are given a quantum implementation $\boldsymbol{U}_{f_\omega}$ of the function f_ω :

$$\boldsymbol{U}_{f_\omega} |x\rangle |y\rangle = |x\rangle |f_\omega(x) \oplus y\rangle\,,$$

   where $|x\rangle$ is n-qubit state and $|y\rangle$ is a single qubit state.

   Here $n = \log_2 N$

In particular,

$$\boldsymbol{U}_{f_\omega}|x\rangle\frac{1}{\sqrt{2}}\big(|0\rangle-|1\rangle\big)=|x\rangle(-1)^{f_\omega(x)}\frac{1}{\sqrt{2}}\big(|0\rangle-|1\rangle\big)$$

Therefore, we will consider $\boldsymbol{U}_\omega$:

$$\boldsymbol{U}_\omega|x\rangle=(-1)^{f_\omega(x)}|x\rangle$$

The operator $\boldsymbol{U}_\omega$ can be written as

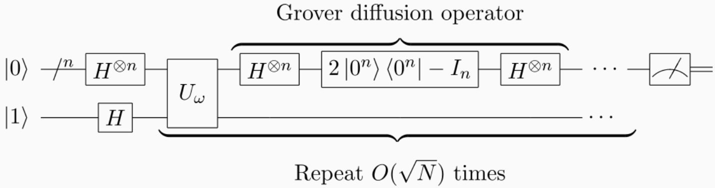
$$\boldsymbol{U}_\omega=\mathbf{1}-2|\omega\rangle\langle\omega|$$

2. Prepare each qubit in the state $|0\rangle$ and apply the Hadamard transformation to each qubit. The result is the state
$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$
3. r times perform the Grover iteration, defined by the unitary transformation
$$\boldsymbol{U}_{\text{grov}} = \boldsymbol{U}_s \boldsymbol{U}_\omega$$
Here
$$\boldsymbol{U}_s = 2|s\rangle\langle s| - \mathbf{1}$$

Since $|s\rangle = \boldsymbol{H}^{(n)}|0\rangle$, it follows

$$\boldsymbol{U}_s = \boldsymbol{H}^{(n)}\big(2|0\rangle\langle 0| - \mathbf{1}\big)\boldsymbol{H}^{(n)}$$

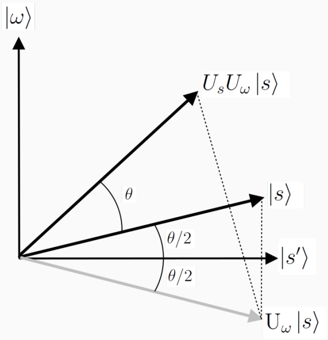
4. Measurement by projecting into the computational basis $\{|x\rangle\}$



Since ω is one of the basis vectors in $|s\rangle$, then

$$|\langle s|\omega\rangle| = \frac{1}{\sqrt{N}} \equiv \sin\frac{\theta}{2}$$

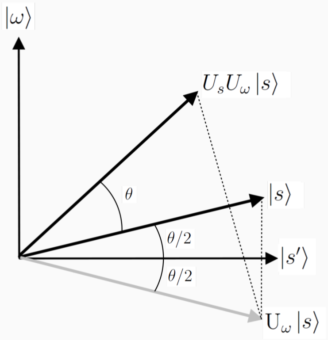
The operator $\boldsymbol{U}_s\boldsymbol{U}_\omega$ of each iteration step rotates the state vector by an angle θ.



Optimal number of iterations should rotate $|s\rangle$ to $|\omega\rangle$, thus

$$r\theta + \frac{\theta}{2} \approx \frac{\pi}{2}$$

We get $r \sim \sqrt{N}$



Discussion

Summary

What makes quantum algorithms interesting is that they *might* be able to solve *some* problems faster than classical algorithms because the quantum superposition and quantum entanglement that quantum algorithms exploit *probably* can't be efficiently simulated on classical computers.

NISQ

J. Preskill, *Quantum Computing in the NISQ era and beyond*, `https://arxiv.org/abs/1801.00862`

Some companies

- IBM: `https://www.ibm.com/quantum-computing/`
- Google: `https://research.google/teams/applied-science/quantum/`
- Rigetti: `https://www.rigetti.com/`
- IonQ: `https://ionq.com/`
- Xanadu: `https://www.xanadu.ai/`

Thank you for your attention!