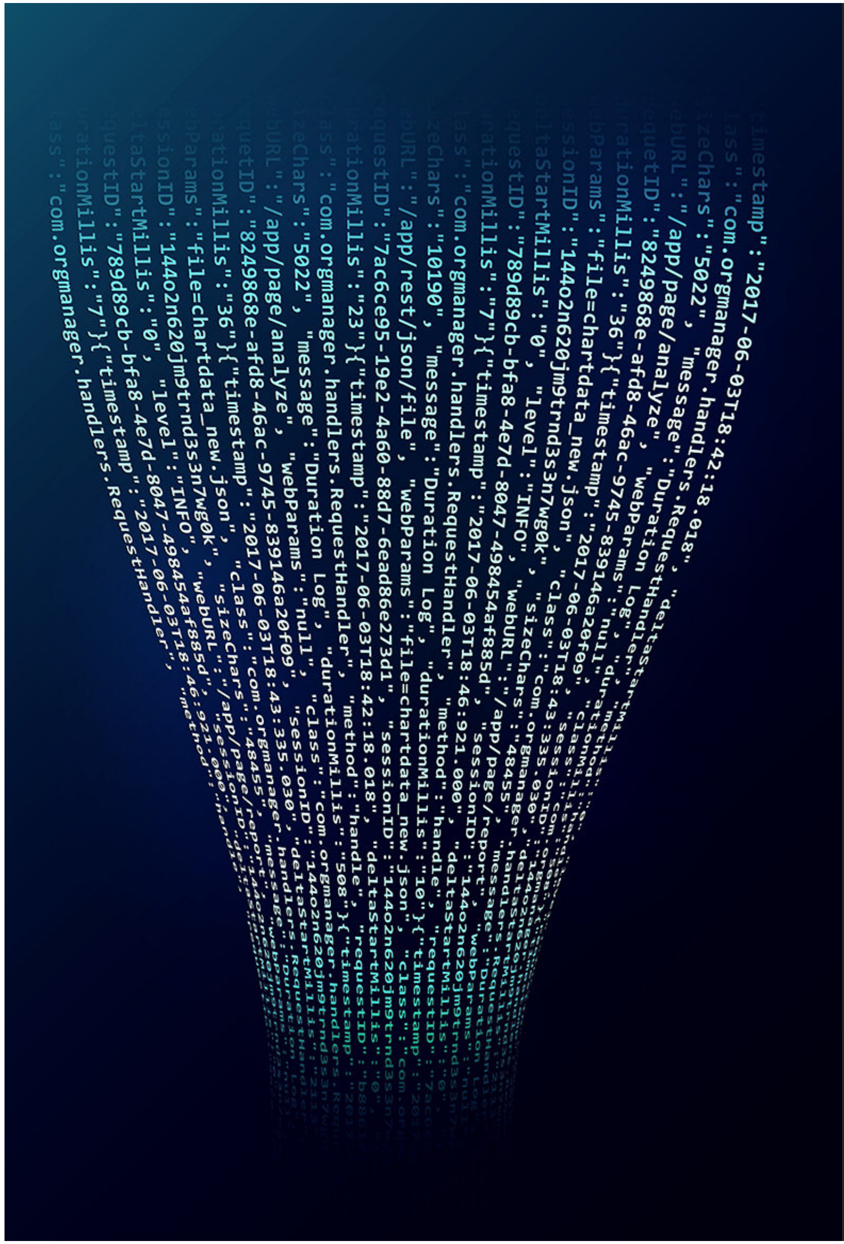


Lecture 6:
Quantum search algorithm

COMP3366
Quantum algorithms & computing architecture

Instructor: Yuxiang Yang

Department of Computer Science, HKU

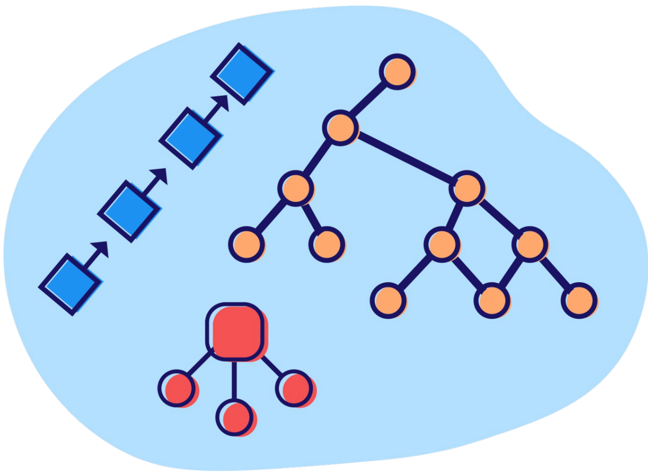
Objectives:

- **[O1] Concepts:** Quantum search algorithm (Grover’s).
- **[O2] Problem solving:** Comprehension of Grover’s algorithm, analysis of its performance, the over-cooking issue.
- **[O3] Algorithm design:** Application of phase kickback trick & oracle model in searching, coding your own Grover using Qiskit (Assignment 3).

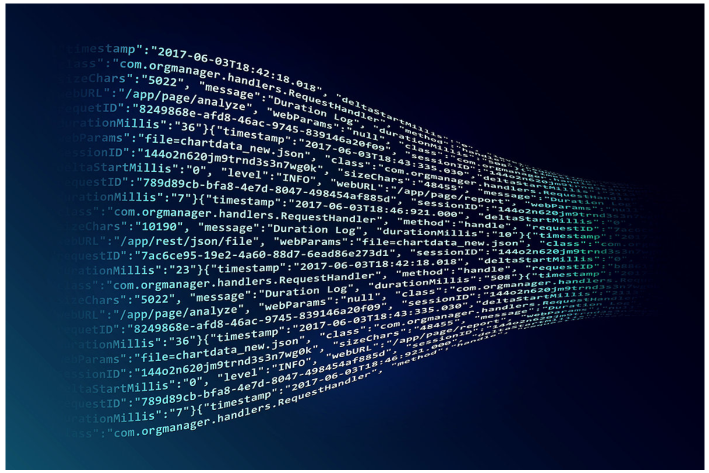
Part I: Search in an unstructured data base

Overview

- In some tasks, a quantum computer brings sub-exponential but still very appealing speedups.
- An example: Search an item in an unstructured database.



Structured



Unstructured

- Grover algorithm: quantum runtime $\approx \sqrt{\text{classical runtime}}$.

Oracle model for unstructured search

- Each query to the database is equivalent to one use of the indicator function: ($1 =$ yes and $0 =$ no):

$$f(x) = \begin{cases} 1 & x = x_0 \\ 0 & x \neq x_0 \end{cases}$$

player

Is the 2nd item the desired one?

No! ($f(2) = 0$)

oracle

$O \to$ apply $(-1)^{f(x)}$ on $|x\rangle$

Classical query complexity

- How many queries to $f(x)$ does a classical algorithm need? $\Omega(N)$
- As the database is totally unstructured, there is not much it can do.
- Worst-case: $N - 1$ times!
- Average case: still $\Omega(N)$!
- This is a bad (hard) task for classical computers, as there is no clever way to accelerate it.
- Remark: In practice, many database has a good structure, and more efficient search algorithms are possible.
- Can we search faster with a quantum computer?

Part II: Grover’s algorithm

Faster search with a quantum computer

- Search an item in an unstructured database of N items.
- Quantum advantage: quantum runtime $\approx \sqrt{\text{classical runtime}}$.

Lov Grover

VOLUME 79, NUMBER 2 PHYSICAL REVIEW LETTERS 14 JULY 1997

Quantum Mechanics Helps in Searching for a Needle in a Haystack

Lov K. Grover*
3C-404A Bell Labs, 600 Mountain Avenue, Murray Hill, New Jersey 07974

(Received 4 December 1996)

Quantum mechanics can speed up a range of search applications over unsorted data. For example, imagine a phone directory containing N names arranged in completely random order. To find someone's phone number with a probability of 50%, any classical algorithm (whether deterministic or probabilistic) will need to access the database a minimum of $0.5N$ times. Quantum mechanical systems can be in a superposition of states and simultaneously examine multiple names. By properly adjusting the phases of various operations, successful computations reinforce each other while others interfere randomly. As a result, the desired phone number can be obtained in only $O(\sqrt{N})$ accesses to the database.
[S0031-9007(97)03564-3]

The quantum oracle for search

- Recall: A quantum oracle for f is a unitary acting on the system $|x\rangle$ and a qubit ancilla $|q\rangle$

$$|x\rangle|q\rangle \mapsto |x\rangle|q \oplus f(x)\rangle$$

- For data base search, we have the indicator function

$$f(x) = \begin{cases} 1 & x = x_0 \\ 0 & x \neq x_0 \end{cases}$$

- Recall the "<u>phase kickback</u>" trick for the oracle: Prepare the ancilla in $|-\rangle$:
  1. If $f(x) = 0 \Rightarrow |x\rangle|-\rangle \mapsto |x\rangle|-\rangle$
  2. If $f(x) = 1 \Rightarrow |x\rangle|-\rangle \mapsto |x\rangle(-|-\rangle)$

  The phase ± 1 can be kicked onto the system as:

$$\underbrace{O = \sum_x (-1)^{f(x)} |x\rangle\langle x|}$$

Quantum search (Grover's) algorithm

- **Input:**
  1. Oracle $O: |x\rangle|q\rangle \mapsto |x\rangle|q \oplus f(x)\rangle$. Here $f(x_0) = 1$ and $f(x) = 0$ for $x \neq x_0$.
- **Output:**
  1. An estimate $\widehat{x_0}$ of x_0.
- **Query complexity**: $O(\sqrt{N})$
- **Circuit depth:** $O(n \cdot \sqrt{N})$

$n \coloneqq \lceil \log N \rceil$ is the number of qubits.

- **Accuracy:** $\widehat{x_0} = x_0$ with probability $\approx 1 - N^{-1}$
- **Procedure:**

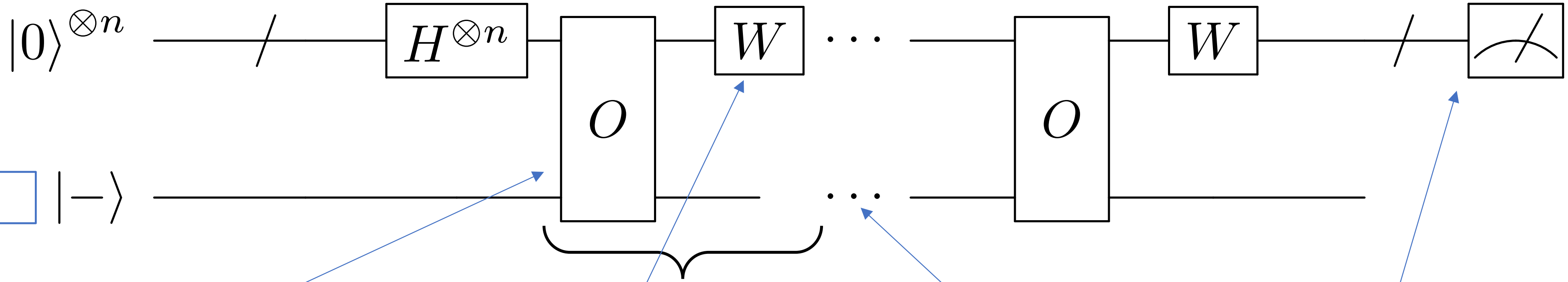
1. Initialize the n-qubit main register and a qubit ancilla in $|0\rangle_S|-\rangle_A$
2. Perform Hadamard transform $H^{\otimes n}$ on S.
3. Repeat the below procedure for $k \approx \left\lceil \frac{\pi\sqrt{N}}{4} \right\rceil$ times:
   1) Apply the oracle O.
   2) Apply $W = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n}$ on S.
4. Measure S in the computational basis and output the outcome as an estimate $\widehat{x_0}$.

The circuit of Grover

one "rotate"

$n(=\log N)$-qubit system

1-qubit ancilla



Acting as a phase gate on the system thanks to the phase kickback trick.

$G = WO$ is the Grover iteration operator.

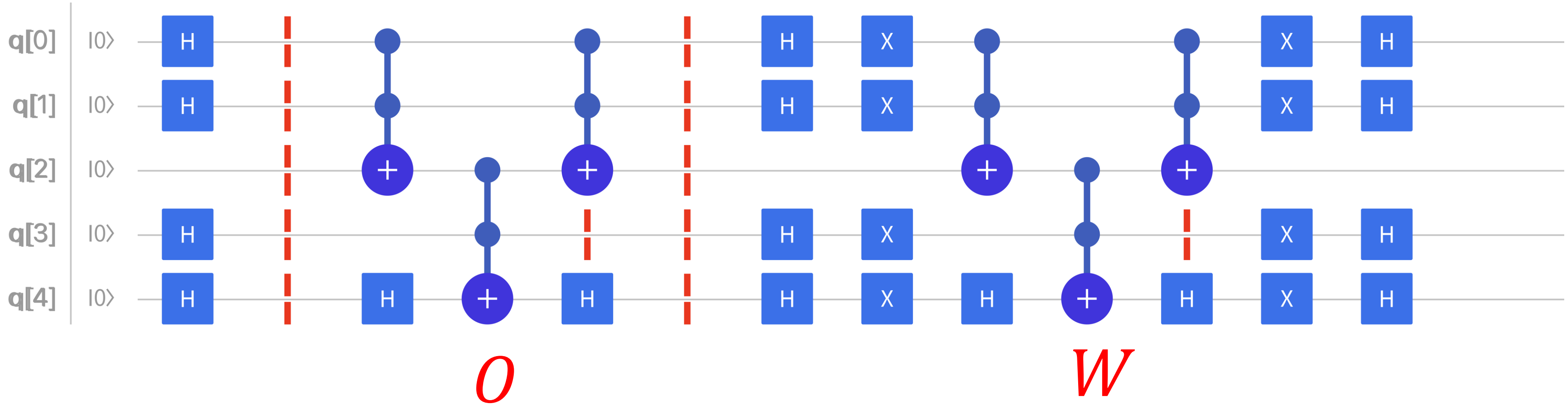
Iterate $k = \Theta(\sqrt{N})$ times

$W = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n}$
(Note: Here $|0\rangle$ denotes the computational zero state for n qubits.)

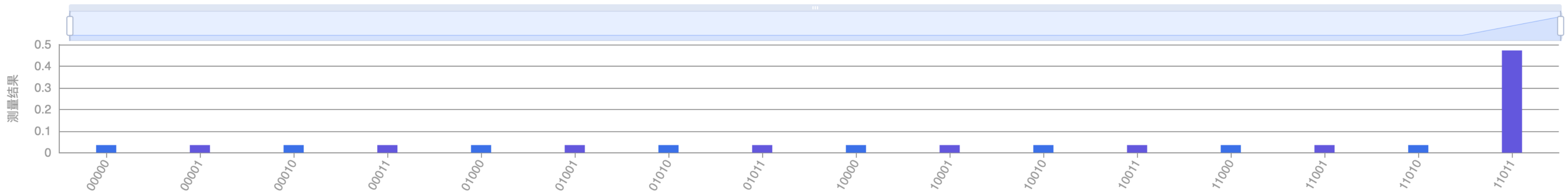
Measurement in the computational basis of n qubits; the outcome is an estimate of the desired item.

Feature: query complexity of Grover = $\Theta(\sqrt{N})$!

Run Grover on OriginQC ($N = 2^4$)



Verify the functionality.
Why 5 qubits? What is the target item?



Part III: Working principle of Grover’s algorithm

Preliminary: Reducing to a 2-D plane

- Initially, the space is N – dimensional:
  There are N items $1, \dots, N$, each assigned a state $|1\rangle, \dots, |N\rangle$
- However, there are actually only 2 types of states: $|x_0\rangle$ and others.
- Observing this symmetry, we can define
  $$|x_0^\perp\rangle \coloneqq \frac{1}{\sqrt{N-1}} \sum_{x \neq x_0} |x\rangle$$
  and Grover's algorithm works within the 2-D space spanned by $\{|x_0\rangle, |x_0^\perp\rangle\}$.
- The initial state of Grover $|e_0\rangle = \frac{1}{\sqrt{N}}\left(|x_0\rangle + \sqrt{N-1}|x_0^\perp\rangle\right)$.
  The Grover's oracle has action $O = (-1)|x_0\rangle\langle x_0| + |x_0^\perp\rangle\langle x_0^\perp|$ in this space.

Step 1: Grover iteration operator as a rotation

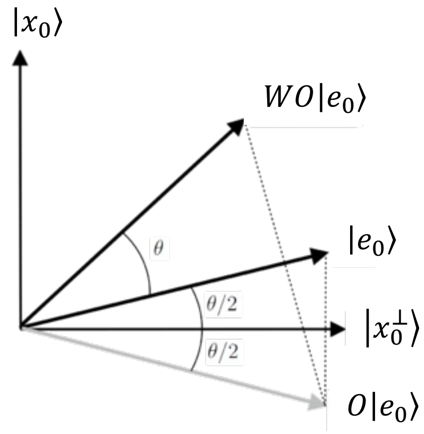
- The spirit of Grover: To rotate $|e_0\rangle$ towards $|x_0\rangle$ in the 2-D plane.
- Problem: How? We don't know x_0!
- A bit of geometry: Within a 2D plane **2 reflections = 1 rotation!**
- The oracle is the 1st reflection:

$$O = \sum_x (-1)^{f(x)} |x\rangle\langle x| = -2|x_0\rangle\langle x_0| + I$$

This is a reflection about $x_0^{\perp}$: $O|x_0^{\perp}\rangle = |x_0^{\perp}\rangle, O|x_0\rangle = -|x_0\rangle$.

- $W = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n}$ is the 2nd reflection.

(Since $H, |0\rangle$ are fixed and do not depend on x_0, we can construct W without the oracle.)



Reflections in the plane spanned by $|x_0\rangle, |e_0\rangle$

Exercise:
Prove that W is a reflection about $|e_0\rangle$.

$|\psi_0\rangle = |0\rangle^{\otimes n}$, $n = \lceil \log_2 N \rceil$ ← 考虑多目标情况.

为简化，我们设 $N=2^n$

$|\psi_1\rangle = H^{\otimes n}|\psi_0\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle$

$|x_0\rangle = \sum_{x\in X_0}\frac{1}{\sqrt{M}}|x\rangle$, $|x_0^\perp\rangle = \frac{1}{\sqrt{N-M}}\sum_{x\notin X_0}|x\rangle$

i.e. $|\psi_1\rangle = \frac{\sqrt{M}}{\sqrt{N}}|x_0\rangle + \frac{\sqrt{N-M}}{\sqrt{N}}|x_0^\perp\rangle = \sin\theta\,|x_0\rangle + \cos\theta\,|x_0^\perp\rangle$

我们考虑 平面 $\{|x_0\rangle, |x_0^\perp\rangle\}$!

$O|x_0\rangle = -|x_0\rangle$, $O|x_0^\perp\rangle = |x_0^\perp\rangle$

i.e. O is "$-Z$" on $\{|x_0\rangle, |x_0^\perp\rangle\}$

i.e. $O \Rightarrow \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ on $\{|x_0\rangle, |x_0^\perp\rangle\}$

$\begin{pmatrix} \sin\theta \\ \cos\theta \end{pmatrix}$ on $\{|x_0\rangle, |x_0^\perp\rangle\}$

$W = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} = 2|\psi_1\rangle\langle\psi_1| - I \Rightarrow 2\begin{pmatrix} \sin^2\theta & \sin\theta\cos\theta \\ \sin\theta\cos\theta & \cos^2\theta \end{pmatrix} - I$

$$= \begin{pmatrix} -\cos 2\theta & \sin 2\theta \\ \sin 2\theta & \cos 2\theta \end{pmatrix}$$

$\therefore$ Grover 算子 $G = WO = \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ -\sin 2\theta & \cos 2\theta \end{pmatrix} = R(-2\theta)$ → 旋转矩阵

i.e. $G|\psi_1\rangle = R(-2\theta)\cdot(\sin\theta|x_0\rangle + \cos\theta|x_0^\perp\rangle) = \begin{pmatrix} \sin 3\theta \\ \cos 3\theta \end{pmatrix}$

经历 k 次 G，得到：

$$\sin((2k+1)\theta)\,|x_0\rangle + \cos((2k+1)\theta)\,|x_0^\perp\rangle$$

此时测量，得到 $|x_0\rangle$ 中计算基的概率：

$$\langle x_0|\sim\ = \sin((2k+1)\theta) \to 1$$

$$\text{when } (2k+1)\theta \to \frac{\pi}{2}.$$

The Grover iteration operator

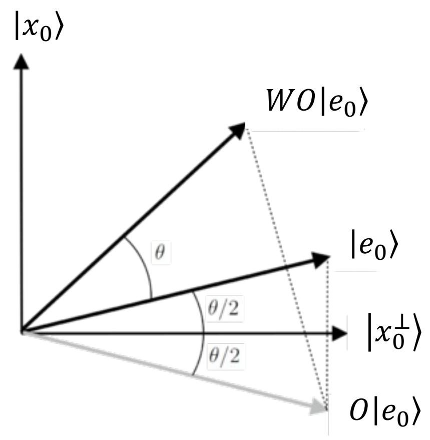
- $G := WO$ is a rotation. What do we know about it?
- Restricting to the plane spanned by $\{|x_0^\perp\rangle, |x_0\rangle\}$, with $|x_0^\perp\rangle := \sqrt{\frac{N-1}{N}} \sum_{x \neq x_0} |x\rangle$, we can rewrite W, O as

$$W = \begin{pmatrix} -\cos\theta & \sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}, O = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

with $\theta := 2 \arcsin \sqrt{1/N}$.

- The Grover iteration is a rotation from $|x_0^\perp\rangle$ to $|x_0\rangle$ by θ:

$$G = WO = -\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$



Exercise:
Verify the above matrix forms of W, O.

Step 2: the optimal number of iterations

- Question: How many times should we apply G?
- The initial state is $|e_0\rangle = \binom{\cos\theta/2}{\sin\theta/2}$. G is a rotation by $\theta = 2\arcsin\sqrt{1/N}$.
- $\Rightarrow$ After k iterations the state becomes $\begin{pmatrix}\cos\left(\frac{1}{2}+k\right)\theta \\ \sin\left(\frac{1}{2}+k\right)\theta\end{pmatrix}$.
- The target is $|x_0\rangle = \binom{0}{1}$. $\Rightarrow$ The minimum k satisfies
$$\boxed{\left(\frac{1}{2}+k\right)\theta \approx \frac{\pi}{2}}$$
- Optimal iteration number for Grover:
We should choose k to be the closest integer k^* to
$$\boxed{\frac{\pi}{4\arcsin\sqrt{1/N}} - \frac{1}{2}}$$

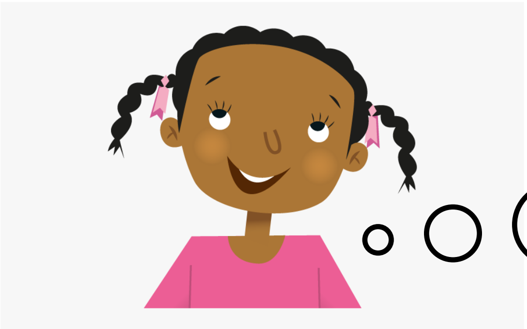
Exercise:
See what happens if we over-rotate (e.g., k being twice the optimal value).

Step 3: success probability

- Query complexity of Grover = $\Theta(\sqrt{N})$, as $k^* = \frac{\pi}{4\arcsin\sqrt{1/N}} - \frac{1}{2} \approx \frac{\pi}{4}\sqrt{N}$.
- Question: What is the probability of success (error)?
- When query complexity $= k$, the state becomes $\begin{pmatrix} \cos\left(\frac{1}{2}+k\right)\theta \\ \sin\left(\frac{1}{2}+k\right)\theta \end{pmatrix}$.
- $\Rightarrow$ the probability of correctly outputting x_0 is $P = \left(\sin\left(\frac{1}{2}+k\right)\theta\right)^2$.
- Since k is the closest integer (gap $\leq 1/2$) to $\frac{\pi}{2\theta} - \frac{1}{2}$ with $\theta = 2\arcsin\sqrt{1/N}$, we have

$$P \geq 1 - \left(\frac{\pi}{2} - \left(\frac{1}{2}+k\right)\theta\right)^2 \geq 1 - \frac{\theta^2}{4} \approx 1 - \frac{1}{N}$$

Optimality of Grover's scaling



Can we do better than $O(\sqrt{N})$?

Maybe another exponential speedup like Shor?

- No!
  Searching an item in an unstructured database of size N requires $\Omega(\sqrt{N})$ queries to the oracle.
- In another word, Grover's algorithm achieves the optimal scaling $\Theta(\sqrt{N})$!
- Proof: See the bonus material.

Part III:
Improvements on Grover

Multi-item search

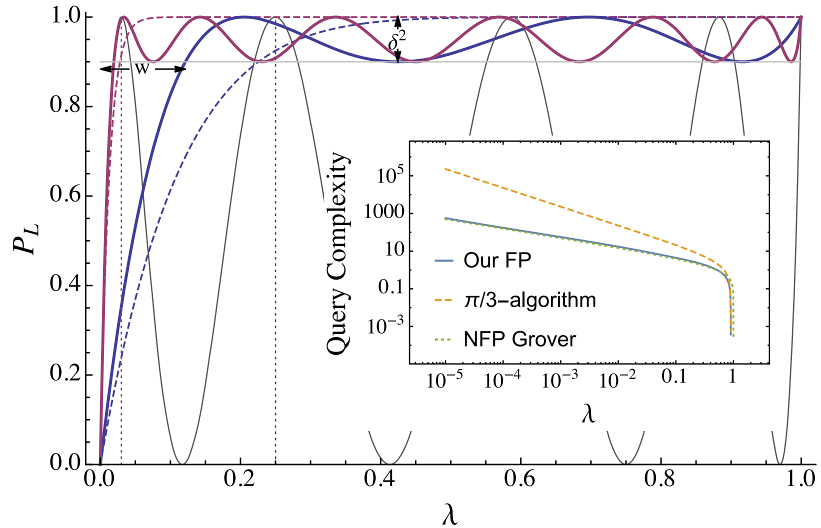
- What if there are $M \geq 1$ items, and the task is to find any one of them amount N items?
- This can be easily remedied, by replacing every N in the original algorithm by N/M!

Exercise:
Verify the above statement.

- The (modified) Grover algorithm yields a random but desired item within $O(\sqrt{N/M})$ queries to the oracle, with success probability $\approx 1 - M/N$!

Remedy for overcooking

- Grover algorithm has a caveat that it requires knowledge of the ratio N/M.
- Otherwise, we have the overcooking problem:
- Remedies:
  - Estimate N/M by sampling before applying Grover.
  - Fixed-point quantum search algorithms (Grover'05,Yoder-Low-Chuang'14)
- Using fixed-point quantum search, the success probability will not vanish under overcooking.
- Question: Can you design a quantum algorithm for estimating N/M?



Success prob. vs $\lambda\left(\sim \frac{M}{N}\right)$ in [Yoder-Low-Chuang'14]

Discussion: Grover vs Shor, HHL

- We have so far learned two categories of quantum algorithms.
- What are their differences and similarities?

Summary

- Grover: a quantum search algorithm in an unstructured database.
- Optimality of Grover[*].
- Improvements on the quantum search algorithm.

Homework

- Review the lecture slides; you may find the review questions in the next slides helpful.
  Try the exercises in the slides and discuss with your classmates.
- Arithmetic details like continued fractions are not important.
  Instead, you should understand how the quantum subroutine works and master the tricks of algorithm design (phase-kick, partial measurement …)
- Attempt Q5 in Assignment 2 and Q1 in Assignment 3.
- Optional: Read p248-255 of *Quantum Computation and Quantum Information* by Nielsen and Chuang.

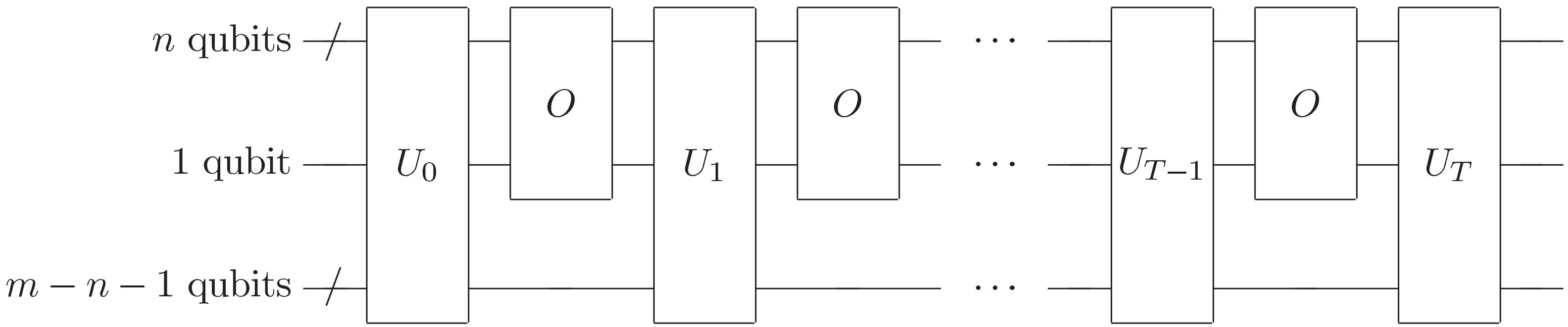
Review questions

- Review the phase kickback trick, which is very useful in quantum algorithm design. Summarize how is it used in Deutsch-Jozsa, QPE (Shor), and Grover.
- In the original version of Grover, find the probability of getting the correct answer if we stop after m interactions. What is the probability when $N = m$?
- What happens to Grover if we don't know N? Are there any solutions?
- Compared to best (existing) classical algorithms, Shor is exponentially faster but Grover is only quadratically faster. What is the cause of this difference?

Bonus content*: Optimality of Grover's algorithm

A general quantum algorithm for searching

- Suppose an arbitrary algorithm makes K queries to the oracle $O_x = -2|x\rangle\langle x| + I$.
- The algorithm does not know x, so it must be of the following general structure:



where $U_1, \ldots, U_K$ are generic unitary operations. They may consist of one or many elementary gates.

- Our goal is to show that $K = \Omega(\sqrt{N})$, and thus no algorithm can do better than Grover in terms of query complexity!

A general quantum algorithm for searching

- Consider the two families of states: for $k = 0,1, \dots, K$ we have

$$|\psi_k^x\rangle = U_k O_x U_{k-1} O_x \cdots U_1 O_x |\psi_0\rangle$$

$$|\psi_k\rangle = U_k U_{k-1} \cdots U_1 |\psi_0\rangle$$

- Remark: $|\psi_k^x\rangle$ is the state after, and $|\psi_k\rangle$ is a reference state obtained by removing the oracles from the algorithm.

- We require that the algorithm finds the correct answer up to a (small) error ϵ when it terminates, for every $x = 1,2, \dots, N$.

  This can be characterized by the following requirement:

$\||a\rangle\| \coloneqq \sqrt{\langle a|a\rangle}$

- Faithfulness: $\||x\rangle - |\psi_K^x\rangle\|^2 \leq \epsilon \ \forall x \in \{1, \dots, N\}$.

Proof strategy

- Consider the overall impact that the oracle has on the system state:

$$D_k := \sum_{x=1}^{N} \| |\psi_k\rangle - |\psi_k^x\rangle \|^2$$

- We shall establish the bound on K (in terms of N), by constructing both a lower bound and an upper bound on D_K.
- Lower bound in terms of N:
  By faithfulness, $|\psi_K^x\rangle \approx |x\rangle$, whereas $|\psi_K\rangle$ is independent of $|x\rangle$ and cannot be close to every $|x\rangle$ at the same time.
- Upper bound in terms of K:
  The two states $|\psi_K\rangle, |\psi_K^x\rangle$ differ only by the oracle action (K times), which is subject to a "speed limit" on how much the oracle can change the state in each iteration.

An upper bound on D_K

- Consider the quantity:

$$D_{k+1} = \sum_x \| U_k |\psi_k\rangle - O_x U_k |\psi_k^x\rangle \|^2$$

$$= \sum_x \| |\psi_k\rangle - O_x |\psi_k^x\rangle \|^2 = \sum_x \| (O_x - I)|\psi_k^x\rangle + (|\psi_k^x\rangle - |\psi_k\rangle) \|^2$$

$$\leq \sum_x \left(\| (O_x - I)|\psi_k^x\rangle \| + \| \, |\psi_k^x\rangle - |\psi_k\rangle \| \right)^2$$

$$O_x = -2|x\rangle\langle x| + I$$

$$= \sum_x 4|\langle x|\psi_k^x\rangle|^2 + 4|\langle x|\psi_k^x\rangle| \cdot \| \, |\psi_k^x\rangle - |\psi_k\rangle \| + \| \, |\psi_k^x\rangle - |\psi_k\rangle \|^2$$

$$\leq 4 + 4\sqrt{D_k} + D_k = \left(\sqrt{D_k} + 2\right)^2$$

An upper bound on D_K

- All together, we get the inductive bound:

$$\sqrt{D_{k+1}} \leq \sqrt{D_k} + 2.$$

- Since $D_0 = 0$, we have the upper bound as

$$D_K \leq 4K^2.$$

A lower bound on D_K

- Faithfulness: $\| |x\rangle - |\psi_k^x\rangle \| \leq \epsilon$ for every $x = 1, \ldots, N$.
- Consider the quantity:

$$D_k \coloneqq \sum_{x=1}^{N} \| |\psi_k\rangle - |\psi_k^x\rangle \|^2$$

- We can lower bound it as

$$D_k \geq \sum_{x=1}^{N} \left(\| |\psi_k\rangle - |x\rangle \| - \| |x\rangle - |\psi_k^x\rangle \| \right)^2$$

$$= \sum_{x=1}^{N} \| |\psi_k\rangle - |x\rangle \|^2 - 2 \| |x\rangle - |\psi_k^x\rangle \| \cdot \| |\psi_k\rangle - |x\rangle \| + \| |x\rangle - |\psi_k^x\rangle \|^2$$

A lower bound on D_K

- We can lower bound D_K as

$$D_K \geq \sum_{x=1}^{N} \| |\psi_K\rangle - |x\rangle\|^2 - 2\| |x\rangle - |\psi_K^x\rangle\| \cdot \| |\psi_K\rangle - |x\rangle\| + \| |x\rangle - |\psi_K^x\rangle\|^2$$

$$T_1 = \sum_x (2 - 2|\langle\psi_K|x\rangle|) \geq 2N - 2\sqrt{N}$$

$$T_2 \leq 2\sqrt{T_3 T_1}$$

$$T_3 = \sum_x \| |x\rangle - |\psi_K^x\rangle\|^2 \leq N \cdot \epsilon$$

Faithfulness: $\| |x\rangle - |\psi_k^x\rangle\|^2 \leq \epsilon$

- Summarizing, we get the lower bound as:

$$D_K \geq \left(\sqrt{T_1} - \sqrt{T_3}\right)^2 \geq \left(2 - 2\sqrt{2\epsilon}\right)N - O(\sqrt{N})$$

Optimality of Grover's scaling

- Combining both bounds, we get:
$$4K^2 \geq D_K \geq \left(2 - 2\sqrt{2\epsilon}\right)N - O\left(\sqrt{N}\right)$$
- Together it implies that $K = \Omega\left(\sqrt{N}\right)$!
- Conclusion:
The $\sqrt{N}$-scaling of search in an unstructured database is optimal!
- Grover's algorithm achieves the optimal scaling $O(\sqrt{N})$!