Quantum algorithms II

Grover's and Shor's algorithms

Jérémie Roland

Université Libre de Bruxelles

Quantum Information & Communication

Outline

1. Grover's algorithm
   - Unstructured search
   - Amplitude amplification
2. Shor's algorithm
   - Quantum Fourier transform
   - Period finding
   - Factoring
   - Quantum phase estimation

Outline

Unstructured search

Definition of the problem

- Input: Black-box access to a function: $f : \{0,1\}^n \to \{0,1\}$
- Promise: $f(x) = \begin{cases} 1 & \text{if } x = m \\ 0 & \text{otherwise} \end{cases}$ (where $m \in \{0,1\}^n$ is unknown)
- Problem: Find m

Notes

- Can be generalized to the case of multiple "marked" elements
- Common misconception about quantum algorithms
  - Can try an exponential number of candidate solutions in parallel
  - If true, could solve this problem very fast
  - Actually, can only get quadratic speedup

2-dimensional subspace

- In the spirit of "trying all candidate solutions in parallel"
  - we first construct the superposition

$$|\psi_0\rangle = H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle$$

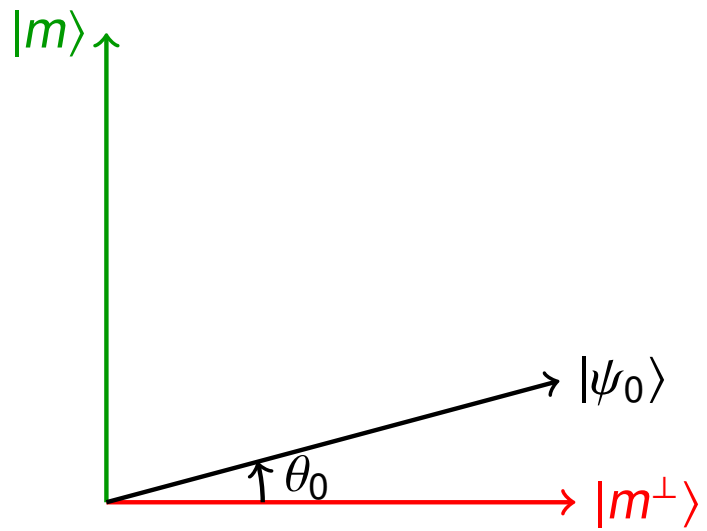
- We are interested in the term $x = m$.
  - Let $|m^\perp\rangle$ correspond to the rest

$$|m^\perp\rangle = \frac{1}{\sqrt{2^n-1}} \sum_{x\neq m} |x\rangle$$

  - Then, $|\psi_0\rangle$ may be rewritten as

$$|\psi_0\rangle = \sin\theta_0|m\rangle + \cos\theta_0|m^\perp\rangle$$

  - where $\theta_0 = \arcsin\langle\psi_0|m\rangle = \arcsin\frac{1}{\sqrt{2^n}}$



- The general idea of Grover's algorithm is to rotate $|\psi_0\rangle$ towards $|m\rangle$

Reflection around unmarked elements

- Recall that

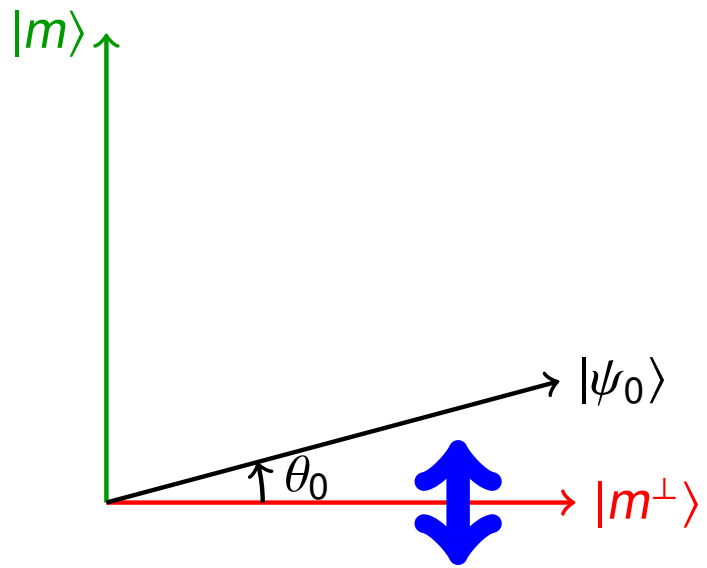
$$f(x) = \begin{cases} 1 & \text{if } x = m \\ 0 & \text{otherwise} \end{cases}$$

- Using phase kickback, we can simulate the operator $U_f : |x\rangle \mapsto (-1)^{f(x)}|x\rangle$
  - In this case, this operator may be rewritten $U_f = I - 2|m\rangle\langle m|$
  - Check:

$$x = m \quad \Rightarrow \quad U_f|m\rangle = |m\rangle - 2|m\rangle\langle m|m\rangle = -|m\rangle = (-1)^{f(m)}|m\rangle$$

$$x \neq m \quad \Rightarrow \quad U_f|x\rangle = |x\rangle - 2|m\rangle\langle m|x\rangle = |x\rangle = (-1)^{f(x)}|x\rangle$$

- In the 2-dim subspace $\{|m\rangle, |m^\perp\rangle\}$
  - $U_f = \mathrm{ref}_{|m^\perp\rangle}$, reflection around $|m^\perp\rangle$



Reflection around the initial state

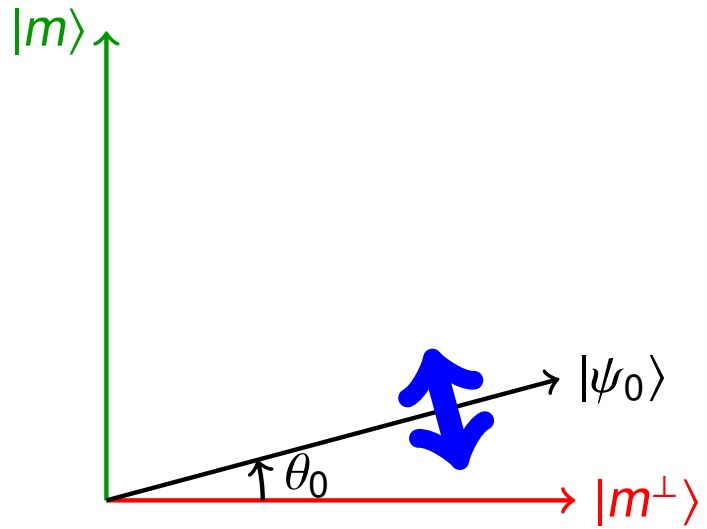
- We can create a circuit that evaluates the function (Exercise!)

$$f_0(x) = \begin{cases} 0 & \text{if } x = 0 \\ 1 & \text{otherwise} \end{cases}$$

- Together with phase kickback, we may simulate $U_0 : |x\rangle \mapsto (-1)^{f_0(x)}|x\rangle$
  - This operator may be rewritten $U_0 = 2|0^n\rangle\langle 0^n| - I$, where $|0^n\rangle = |0\rangle^{\otimes n}$
- Since $H^{\otimes n}|0^n\rangle = |\psi_0\rangle$,

$$\begin{aligned} H^{\otimes n} U_0 H^{\otimes n} &= H^{\otimes n}(2|0^n\rangle\langle 0^n| - I)H^{\otimes n} = 2H^{\otimes n}|0^n\rangle\langle 0^n|H^{\otimes n} - I \\ &= 2|\psi_0\rangle\langle\psi_0| - I \end{aligned}$$

- In the 2-dim subspace $\{|m\rangle, |m^\perp\rangle\}$
  - $H^{\otimes n} U_0 H^{\otimes n} = \mathrm{ref}_{|\psi_0\rangle}$, reflection around $|\psi_0\rangle$

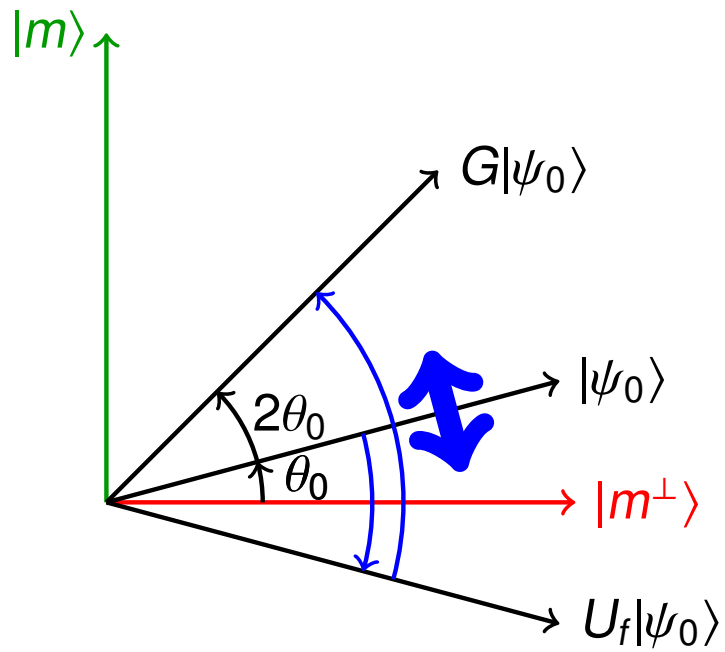


Combining the reflections

- Recall: In the 2-dim subspace $\{|m\rangle, |m^\perp\rangle\}$
  - $U_f = \mathrm{ref}_{|m^\perp\rangle}$
  - $H^{\otimes n} U_0 H^{\otimes n} = \mathrm{ref}_{|\psi_0\rangle}$
- The product of the reflections $G = \mathrm{ref}_{|\psi_0\rangle} \cdot \mathrm{ref}_{|m^\perp\rangle}$ ("Grover iteration") is a
  - rotation of angle $2\theta_0$
  - (where θ is the angle between $|m^\perp\rangle$ and $|\psi_0\rangle$)

- Example

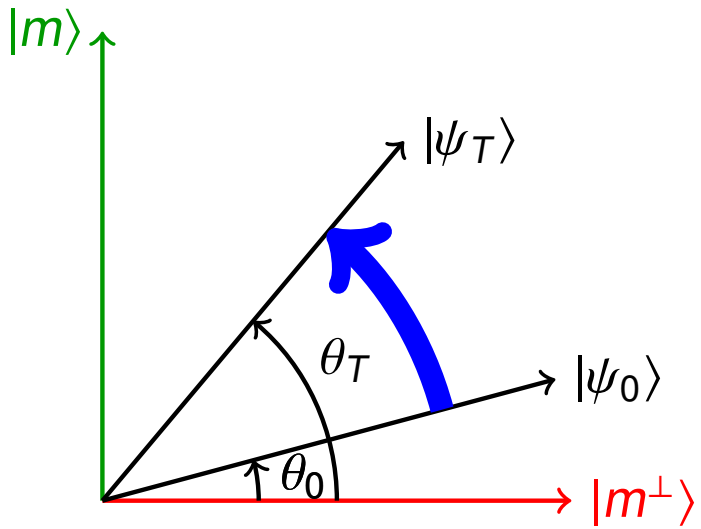
$$|\psi_0\rangle \xrightarrow{U_f} U_f|\psi_0\rangle \xrightarrow{H^{\otimes n} U_0 H^{\otimes n}} G|\psi_0\rangle$$



Grover's algorithm [Gro96]

- Starting from
$$|\psi_0\rangle = \sin\theta_0|m\rangle + \cos\theta_0|m^\perp\rangle$$
- and repeating T times the Grover iteration, we have
$$G^T|\psi_0\rangle = \sin\theta_T|m\rangle + \cos\theta_T|m^\perp\rangle$$
  - where $\theta_T = (2T+1)\theta_0$

- In order to obtain $|m\rangle$
  - we need $\theta_T = \frac{\pi}{2}$
  - and therefore (for large n)
$$\begin{aligned} T &\approx \frac{\pi}{4\theta_0} \\ &= \frac{\pi}{4\arcsin\frac{1}{\sqrt{2^n}}} \\ &\approx \frac{\pi}{4}\sqrt{2^n} \end{aligned}$$



Query complexity of the problem

Quantum query complexity

- Grover's algorithm solves the problem with $O(\sqrt{2^n})$ queries

Classical query complexity

- Recall: the problem is to find the unique m such that $f(m) = 1$
  - among $\{0,1\}^n$, that is a set of size 2^n
- In the worst case, a classical algorithm must query the whole set
  - Classical query complexity: $\Omega(2^n)$

Notes

- Grover's algorithm provides a quadratic speedup
- We can show that this is the best we can obtain for this problem

Outline

Tool: Amplitude amplification [BHMT02]

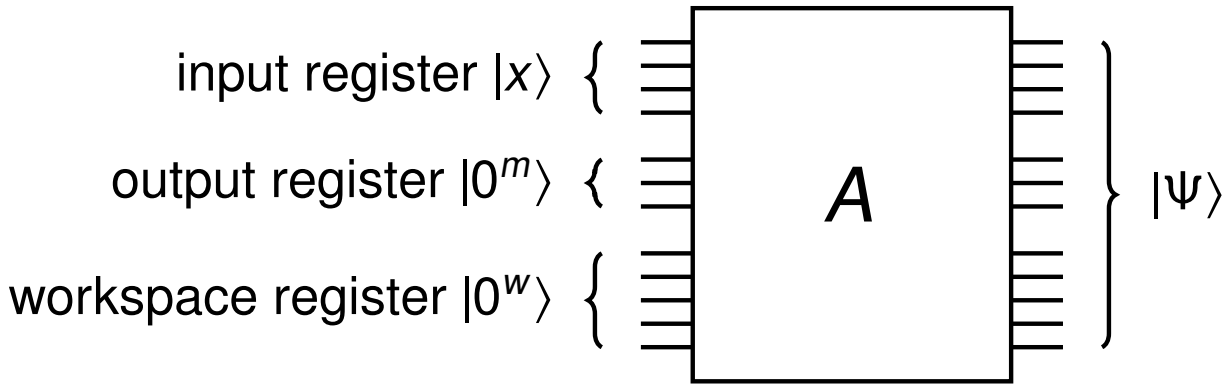
- Let $g : \{0,1\}^n \to \{0,1\}^m$ be a function we would like to compute
- Suppose we have
  - an algorithm computing g with success probability $p \ll 1$
  - a procedure to check if the output is correct or not
- How to construct an algorithm for g with success probability ≈ 1?
  - Classical solution: repeat the algorithm $\Theta(1/p)$ times until the output is correct
  - Quantum solution: use amplitude amplification, which only requires $\Theta(1/\sqrt{p})$ calls to the algorithm.

Tool: Amplitude amplification

Using $\Theta(1/\sqrt{p})$ calls to an algorithm with success probability p, one can construct an algorithm with success probability close to 1.

Quantum version of the original algorithm

- We can assume that the original algorithm is quantum
  - Recall that every classical algorithm can be turned into a quantum algorithm
  - For probabilistic algorithms, we can produce random bits by measuring $\frac{1}{\sqrt{2}}[|0\rangle + |1\rangle]$
- Let A be the unitary implemented by the quantum circuit
- Let $|\Psi\rangle$ be the final state (before any final measurement)



- Since the algorithm computes $g(x)$ with success probability p, we can write

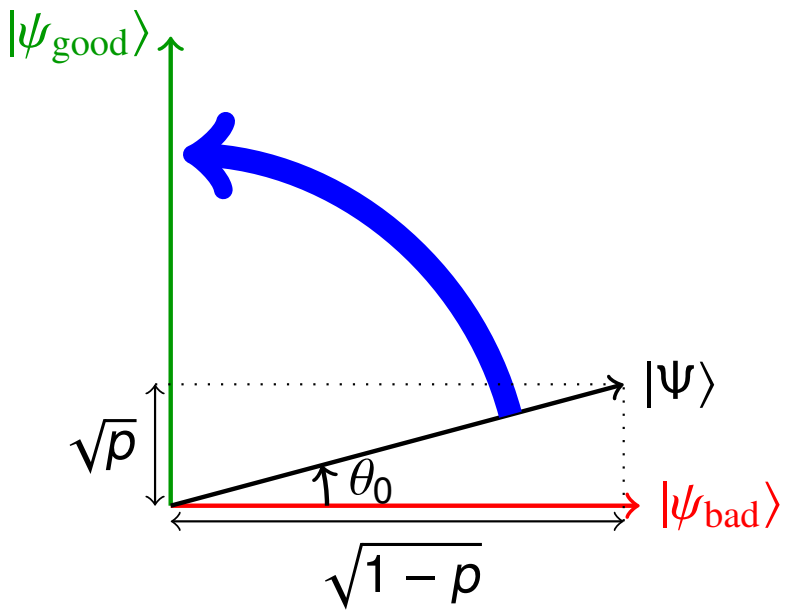
$$|\Psi\rangle = A|x\rangle|0^m\rangle|0^w\rangle = \underbrace{\sqrt{p}|x\rangle|g(x)\rangle|\psi_w^{g(x)}\rangle}_{|\psi_{\text{good}}\rangle} + \underbrace{\sqrt{1-p}\sum_{y\neq g(x)}|x\rangle|y\rangle|\psi_w^y\rangle}_{|\psi_{\text{bad}}\rangle}$$

2-dimensional subspace

$$|\Psi\rangle = A|x\rangle|0^m\rangle|0^w\rangle = \sqrt{p}|x\rangle|g(x)\rangle|\psi_W^{g(x)}\rangle + \sqrt{1-p}\sum_{y\neq g(x)}|x\rangle|y\rangle|\psi_W^y\rangle$$

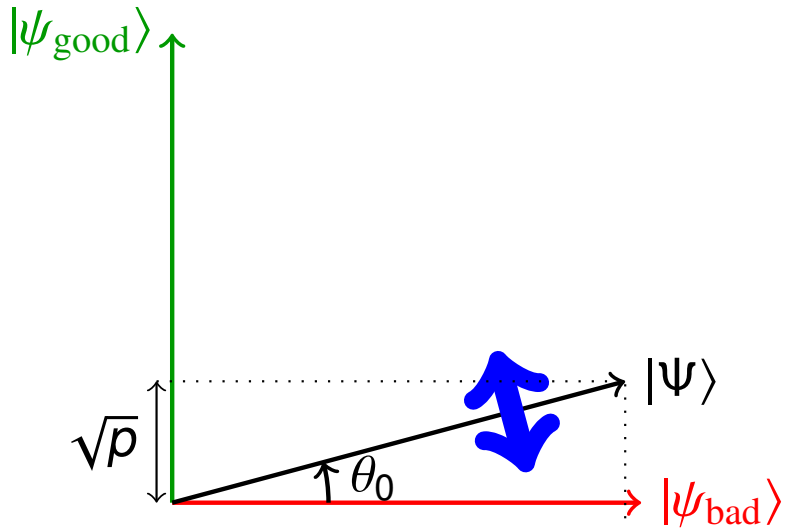
$$= \sqrt{p}|\psi_{\text{good}}\rangle + \sqrt{1-p}|\psi_{\text{bad}}\rangle$$

- We consider the two-dimensional subspace spanned by $|\psi_{\text{good}}\rangle$ and $|\psi_{\text{bad}}\rangle$



- We want to rotate $|\Psi\rangle$ to $|\psi_{\text{good}}\rangle$
  - $\theta_0 = \arcsin\sqrt{p}$
- Idea: Use two reflections as in Grover's algorithm!

Reflection around the initial state



- Recall that

$$|\Psi\rangle = A|x\rangle|0^m\rangle|0^w\rangle$$

- Just as for Grover, we can simulate by phase kickback

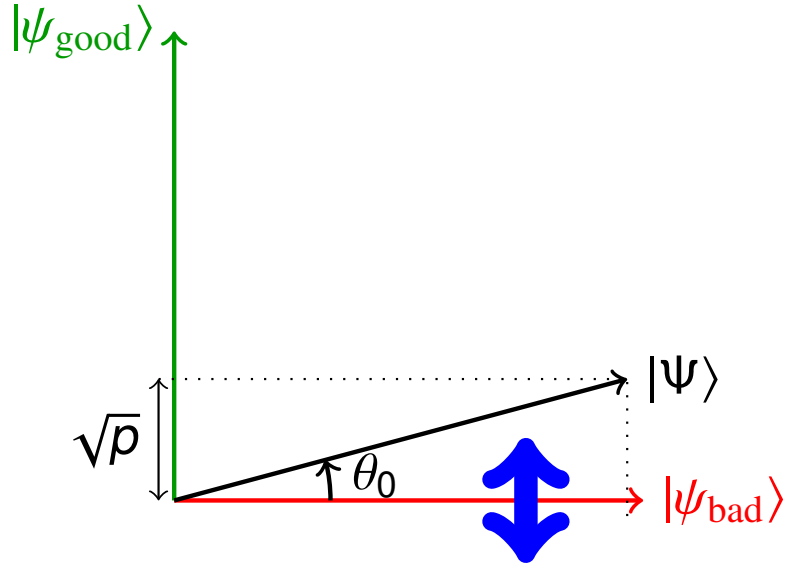
$$U_0 = 2|x\rangle|0^m\rangle|0^w\rangle\langle x|\langle 0^m|\langle 0^w| - I$$

- and therefore simulate $\text{ref}_{|\Psi\rangle}$

$$\begin{aligned} AU_0A^\dagger &= A(2|x\rangle|0^m\rangle|0^w\rangle\langle x|\langle 0^m|\langle 0^w| - I)A^\dagger \\ &= 2|\Psi\rangle\langle\Psi| - I \end{aligned}$$

  - This requires to run the algorithm back ($A^\dagger$) and forth (A)!

Reflection around the bad state



- Recall we have access to a procedure that checks the output of the algorithm
  - we can evaluate the function

$$g : \{0,1\}^n \times \{0,1\}^m \to \{0,1\} : (x,y) \mapsto \begin{cases} 1 & \text{if } y = f(x) \\ 0 & \text{otherwise} \end{cases}$$

- Using phase kickback (again!), we can simulate

$$U_g : |x\rangle|y\rangle \mapsto (-1)^{g(x,y)}|x\rangle|y\rangle$$

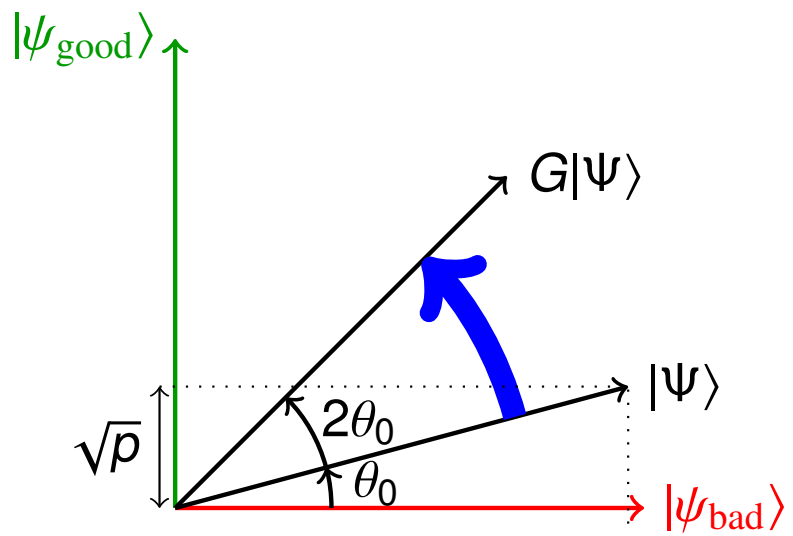
- In particular, in the subspace spanned by $|\psi_{\text{good}}\rangle$ and $|\psi_{\text{bad}}\rangle$

$$U_g|\psi_{\text{good}}\rangle = -|\psi_{\text{good}}\rangle \qquad U_g|\psi_{\text{bad}}\rangle = |\psi_{\text{bad}}\rangle$$

  - This is $\text{ref}_{|\psi_{\text{bad}}\rangle}$, the reflection we need!

Wrapping up

- Using one call to A, we prepare
$$|\Psi\rangle = A|x\rangle|0^m\rangle|0^w\rangle$$
- Combining the two reflections
$$G = AU_0A^\dagger U_g = \mathrm{ref}_{|\Psi\rangle} \cdot \mathrm{ref}_{|\psi_{\mathrm{bad}}\rangle}$$
  - we obtain a rotation of angle
$$2\theta_0 = 2\arcsin\frac{1}{\sqrt{p}}$$
$$\approx 2\sqrt{p}$$
- Repeating T times G, where
$$T \approx \frac{\pi}{4\sqrt{p}}$$
  - we rotate $|\Psi\rangle$ to $|\psi_{\mathrm{good}}\rangle$



Outline

1. Grover's algorithm
   - Unstructured search
   - Amplitude amplification
2. Shor's algorithm
   - Quantum Fourier transform
   - Period finding
   - Factoring
   - Quantum phase estimation

Fourier transform over $\mathbb{Z}_N$

- An n-bit string $x = x_{n-1}x_{n-2}\ldots x_1x_0$ can be viewed as
  - an element of $\{0,1\}^n$

$$x = (x_{n-1}, x_{n-2}, \ldots, x_1, x_0)$$

  - an integer $0 \leq x \leq N$, where $N = 2^n$, that is, an element of $\mathbb{Z}_N$

$$x = 2^{n-1}x_{n-1} + 2^{n-2}x_{n-2} + \ldots + 2x_1 + x_0$$

- Recall: the Hadamard transform $H^{\otimes n}$ implements the quantum Fourier transform over $\{0,1\}^n$

$$\sum_{x\in\{0,1\}^n} \psi(x)|x\rangle \xrightarrow{H^{\otimes n}} \sum_{y\in\{0,1\}^n} \hat{\psi}(y)|y\rangle$$

  - where

$$\hat{\psi}(y) = \frac{1}{\sqrt{2^n}} \sum_{y\in\{0,1\}^n} (-1)^{x\cdot y}\psi(x)$$

$x\cdot y$: $\sum_i x_i y_i$

  - is the Fourier transform over $\{0,1\}^n$

Fourier transform over $\mathbb{Z}_N$

- An n-bit string $x = x_{n-1}x_{n-2}\ldots x_1x_0$ can be viewed as
  - an element of $\{0,1\}^n$

$$x = (x_{n-1}, x_{n-2}, \ldots, x_1, x_0)$$

  - an integer $0 \leq x \leq N$, where $N = 2^n$, that is, an element of $\mathbb{Z}_N$

$$x = 2^{n-1}x_{n-1} + 2^{n-2}x_{n-2} + \ldots + 2x_1 + x_0$$

- Question: can we also implement the quantum Fourier transform over $\mathbb{Z}_N$?

$$\sum_{x\in\mathbb{Z}_N} \psi(x)|x\rangle \xrightarrow{QFT_n} \sum_{y\in\mathbb{Z}_N} \hat{\psi}(y)|y\rangle$$

  - where

usual product

$$\hat{\psi}(y) = \frac{1}{\sqrt{2^n}} \sum_{y\in\mathbb{Z}_N} \omega^{xy}\psi(x)$$

  - is the Fourier transform over $\mathbb{Z}_N$, with $\omega = e^{\frac{2\pi i}{N}}$

Quantum Fourier transform (QFT) over $\mathbb{Z}_N$

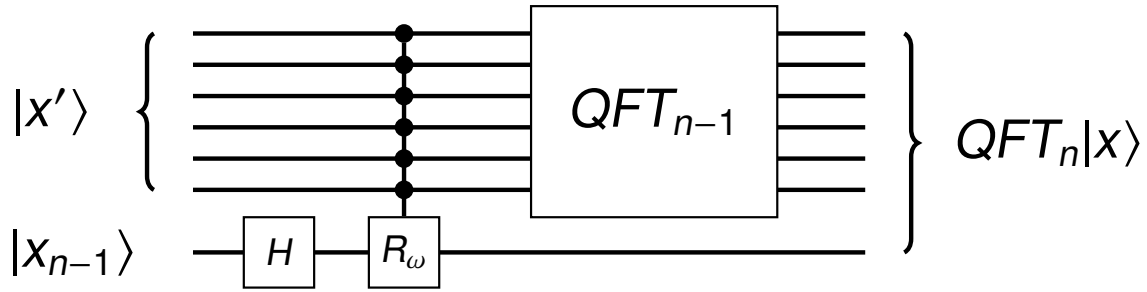
- Let us decompose x and y (integers in binary notation) as follows

$$x = x_{n-1}\underbrace{x_{n-2}\dots x_1 x_0}_{x'} = 2^{n-1}x_{n-1} + x' \qquad y = \underbrace{y_{n-1}y_{n-2}\dots y_1}_{y'}y_0 = 2y' + y_0$$

- By definition QFT_n acts on computational basis states $|x\rangle$ as

$$\begin{aligned}
QFT_n|x\rangle &= \tfrac{1}{\sqrt{2^n}} \sum_{y\in\mathbb{Z}_N} \omega^{xy}|y\rangle \\
&= \tfrac{1}{\sqrt{2^n}} \left[\sum_{y'\in\mathbb{Z}_{N/2}} \omega^{2xy'}|y'0\rangle + \sum_{y'\in\mathbb{Z}_{N/2}} \omega^{x(2y'+1)}|y'1\rangle \right] \\
&= \tfrac{1}{\sqrt{2^n}} \sum_{y'\in\mathbb{Z}_{N/2}} \omega^{2xy'}|y'\rangle \otimes \Big[|0\rangle + \omega^{x}|1\rangle\Big] \\
&= \tfrac{1}{\sqrt{2^n}} \sum_{y'\in\mathbb{Z}_{N/2}} \omega^{2x'y'}|y'\rangle \otimes \Big[|0\rangle + (-1)^{x_{n-1}}\omega^{x'}|1\rangle\Big] \\
&= QFT_{n-1}|x'\rangle \otimes \tfrac{1}{\sqrt{2}}\Big[|0\rangle + (-1)^{x_{n-1}}\omega^{x'}|1\rangle\Big]
\end{aligned}$$

QFT circuit



$$QFT_n|x\rangle = QFT_{n-1}|x'\rangle \otimes \frac{1}{\sqrt{2}}\Big[|0\rangle + (-1)^{x_{n-1}}\omega^{x'}|1\rangle\Big]$$

- In order to create the last qubit state
  - we first apply H

$$|x_{n-1}\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}\Big[|0\rangle + (-1)^{x_{n-1}}|1\rangle\Big]$$

  - followed by the controlled-phase gate R_ω, acting as

$$|x'\rangle|0\rangle \xrightarrow{R_\omega} |x'\rangle|0\rangle \qquad\qquad |x'\rangle|1\rangle \xrightarrow{R_\omega} \omega^{x'}|x'\rangle|1\rangle$$

- It then remains to apply QFT_{n-1} on $|x'\rangle$ by using the same construction recursively

QFT: final remarks

$$QFT_n|x\rangle = QFT_{n-1}|x'\rangle \otimes \frac{1}{\sqrt{2}}\Big[|0\rangle + (-1)^{x_{n-1}}\omega^{x'}|1\rangle\Big]$$

Notes

- After n recursive steps, we obtain $QFT_1 = H$
- Each step introduces $O(n)$ gates
  - Total complexity: $O(n^2)$
  - This is exponentially faster than the classical Fast Fourier Transform (FFT)
- Note that the final state is not entangled (product state)

Tool: Quantum Fourier transform

The quantum Fourier transform over $\mathbb{Z}_N$ (QFT) can be implemented using $O(n^2)$ gates (where $N = 2^n$)

Outline

Period finding

Definition of the problem

- Input: Black-box access to a function: $f : \mathbb{Z}_N \to \mathbb{Z}_N$
- Promise: f is periodic

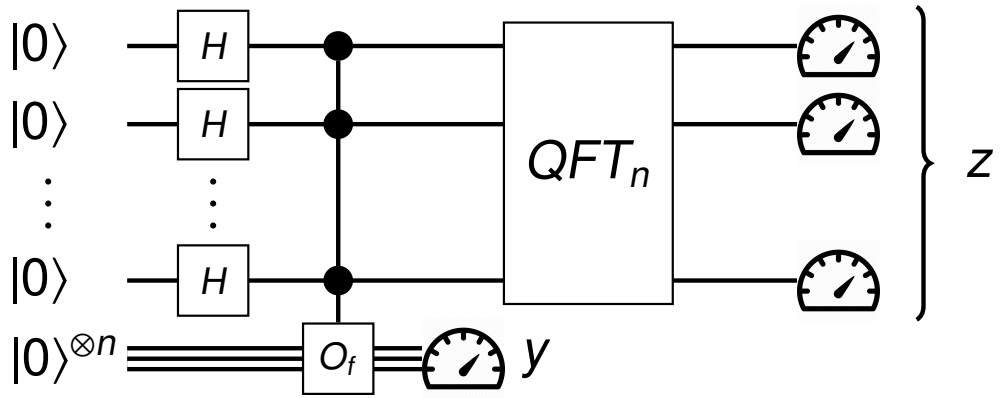
$$f(x) = f(x') \qquad \Leftrightarrow \qquad \exists k : x' = x + ka \mod N$$

- Problem: Find the period a

Notes

- For simplicity, we assume that N is a power of 2 ($N = 2^n$)
  - Otherwise, we round up N to the nearest power of 2
  - f is not perfectly periodic anymore, but algorithm still works w.h.p.

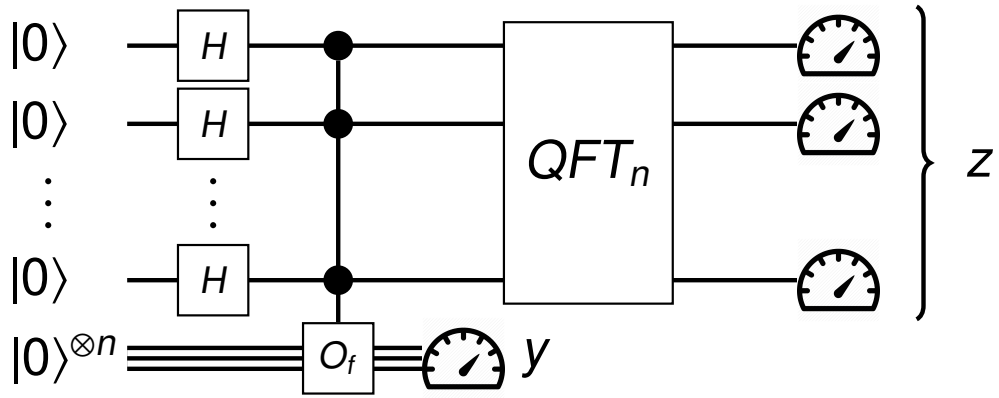
Period finding algorithm [Sho94]



General idea

- We follow the same approach as for Simon's algorithm
  - Also a period finding problem, over $\mathbb{Z}_N$ instead of $\{0,1\}^n$
  - Therefore, use QFT_n instead of $H^{\otimes n}$
- Why would it work?
  - Fourier transform of periodic function is peaked at harmonics of the frequency

Period finding algorithm: analysis



Analysis

$$|0\rangle^{\otimes n}|0\rangle^{\otimes n} \xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{N}} \sum_{x\in\mathbb{Z}_N} |x\rangle|0\rangle^{\otimes n} \xrightarrow{O_f} \frac{1}{\sqrt{N}} \sum_{x\in\mathbb{Z}_N} |x\rangle|f(x)\rangle$$

- If the outcome of the first measurement is y, we have prepared

$$|\psi_y\rangle = \frac{1}{\sqrt{|f^{-1}(y)|}} \sum_{x:f(x)=y} |x\rangle = \sqrt{\frac{a}{N}} \sum_{k=0}^{\frac{N}{a}-1} |x_0 + ka\rangle$$

  - where x_0 is such that $f(x_0) = y$

Period finding algorithm: analysis (cont'd)

- We have prepared the state

$$|\psi_y\rangle = \sqrt{\frac{a}{N}} \sum_{k=0}^{\frac{N}{a}-1} |x_0 + ka\rangle = \sum_{x \in \mathbb{Z}_N} \psi(x)|x\rangle$$

  - where

$$\psi(x) = \begin{cases} \sqrt{\frac{a}{N}} & \text{if } x = x_0 + ka \\ 0 & \text{otherwise} \end{cases}$$

- The quantum Fourier Transform acts on this state as

$$\sum_{x \in \mathbb{Z}_N} \psi(x)|x\rangle \xrightarrow{QFT_n} \sum_{z \in \mathbb{Z}_N} \hat{\psi}(z)|z\rangle$$

  - where

Recall : $\omega = e^{\frac{2i\pi}{N}}$

$$\hat{\psi}(z) = \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_N} \psi(x)\omega^{xz} = \frac{1}{\sqrt{N}} \sum_{k=0}^{\frac{N}{a}-1} \sqrt{\frac{a}{N}} \omega^{(x_0+ka)z} = \frac{\sqrt{a}}{N} \omega^{x_0 z} \sum_{k=0}^{\frac{N}{a}-1} \omega^{kaz}$$

Period finding algorithm: analysis (cont'd 2)

$$\hat{\psi}(z) = \frac{\sqrt{a}}{N}\omega^{x_0 z}\sum_{k=0}^{\frac{N}{a}-1}\omega^{kaz}$$

- We have a geometric sum Recall : $\omega = e^{\frac{2i\pi}{N}} \Rightarrow \omega^N = 1$

$$\sum_{k=0}^{\frac{N}{a}-1}\omega^{kaz} = 1 + \omega^{az} + \omega^{2az} + \ldots + \omega^{(\frac{N}{a}-1)az}$$

$$= \begin{cases} \frac{N}{a} & \text{if } \omega^{az} = 1 \Leftrightarrow z = \frac{jN}{a} \\ \frac{1-\omega^{az\frac{N}{a}}}{1-\omega^{az}} = \frac{1-\omega^{Nz}}{1-\omega^{az}} = 0 & \text{otherwise} \end{cases}$$

- The final state we measure is therefore

$$\sum_{z\in\mathbb{Z}_N}\hat{\psi}(z)|z\rangle = \frac{1}{\sqrt{a}}\sum_{j=0}^{a-1}\omega^{\frac{x_0 jN}{a}}|\frac{jN}{a}\rangle$$

- The measurement outcome z is distributed uniformly in

$$\{0, \frac{N}{a}, \frac{2N}{a}, \frac{3N}{a}, \ldots, \frac{(a-1)N}{a}\}$$

  - These are the harmonics of the frequency $\frac{N}{a}$!

Period finding algorithm: analysis (cont'd 3)

- Can we obtain a from one value $\frac{jN}{a}$?
  - No: we know N but not j (which harmonic)
- Just as for Simon's algorithm, we can repeat and obtain k harmonics
  - They are all multiples of $\frac{N}{a}$
  - If we are lucky, their greatest common divisor (GCD) could be $\frac{N}{a}$
  - Note: the GCD can be computed efficiently using Euclid's algorithm [EucBC]
- How lucky should we be?
  - We obtain another GCD if all harmonics are multiples of $\frac{jN}{a}$ for some $j \geq 2$.

$$\Pr[1 \text{ sample multiple of } \tfrac{jN}{a}] \leq \tfrac{1}{j} \leq \tfrac{1}{2}$$

$$\Pr[\text{k samples multiple of } \tfrac{jN}{a}] \leq \tfrac{1}{2^k}$$

$$\Pr[\exists j : \text{k samples multiple of } \tfrac{jN}{a}] \leq \tfrac{|\#j|}{2^k} \leq \tfrac{a}{2^k} \leq \tfrac{N}{2^k}$$

  - We therefore obtain GCD=$\frac{N}{a}$ w.h.p. by taking $k = \Theta(n)$ samples
- The query complexity of the algorithm is $O(n)$ and the time complexity $O(n^3)$
  - Classically, the complexity of period finding is exponential.

Outline

1. Grover's algorithm
   - Unstructured search
   - Amplitude amplification
2. Shor's algorithm
   - Quantum Fourier transform
   - Period finding
   - Factoring
   - Quantum phase estimation

Factoring

Definition of the problem

- Input: an integer N
- Promise: $N = p_1 \cdot p_2$, where p_1 and p_2 are prime
- Problem: Find the factors p_1 and p_2

Notes

- This is **not** an oracle problem
  - (no black-box, N is fully known to start with)
- General idea: **classical** reduction to period finding
  - In the period finding algorithm, the black-box will be replaced by an explicit circuit computing a function

Shor's algorithm [Sho94]

- Pick an integer $c < N$ uniformly at random
- Compute $GCD(c, N)$ (using Euclid's algorithm)
- If $GCD(c, N) \neq 1$
  - $GCD(c, N)$=p_1 or p_2 $\Rightarrow$ Finished!
- Otherwise
  - The function $f_{N,c}(x) = c^x \mod N$ is a periodic function
  - The period a is the smallest integer c such that $c^a \mod N = 1$ (order of c)
- Compute the period of $f_{N,c}$ using the period finding algorithm
  - Note that $f_{N,c}(x) = c^x \mod N$ can be computed efficiently by "repeated squaring"
- Suppose that a is even (1). Then we have

$$c^a \mod N = 1 \quad \Leftrightarrow \quad c^a - 1 = 0 \mod N$$
$$\Leftrightarrow \quad (c^{a/2} - 1)(c^{a/2} + 1) = 0 \mod N$$

  - $(c^{a/2} - 1)(c^{a/2} + 1)$ is a multiple of $N = p_1 \cdot p_2$

$$(c^{a/2} - 1)(c^{a/2} + 1) = q \cdot p_1 \cdot p_2$$

Shor's algorithm (cont'd) [Sho94]

$$(c^{a/2} - 1)(c^{a/2} + 1) = q \cdot p_1 \cdot p_2$$

- $c^{a/2} - 1$ cannot be a multiple of N
  - Otherwise $c^{a/2} = 1 \mod N$, meaning that the order of c is at most $a/2$
- Assume that $c^{a/2} + 1$ is not a multiple of N either (2)
  - N must have a common factor with each of $(c^{a/2} - 1)$ and $(c^{a/2} + 1)$
  - $GCD(N, c^{a/2} - 1)$ and $GCD(N, c^{a/2} + 1)$ are equal to p_1 and p_2

Discussion

- The algorithm works if
  1. The order a of c is even
  2. $c^{a/2} + 1$ is not a multiple of N
- It can be shown that for a randomly chosen $c < N$, this happens with probability at least $1/2$
  - Pick random c until finding a good value

Outline

1. Grover's algorithm
   - Unstructured search
   - Amplitude amplification
2. Shor's algorithm
   - Quantum Fourier transform
   - Period finding
   - Factoring
   - Quantum phase estimation

Tool: quantum phase estimation

The problem

- Consider a unitary operator U
- Suppose we are given a state $|v\rangle$, and promised that it is an eigenstate of U
- How to estimate the corresponding eigenvalue $e^{2i\pi\phi}$?

$$U|v\rangle = e^{2i\pi\phi}|v\rangle$$

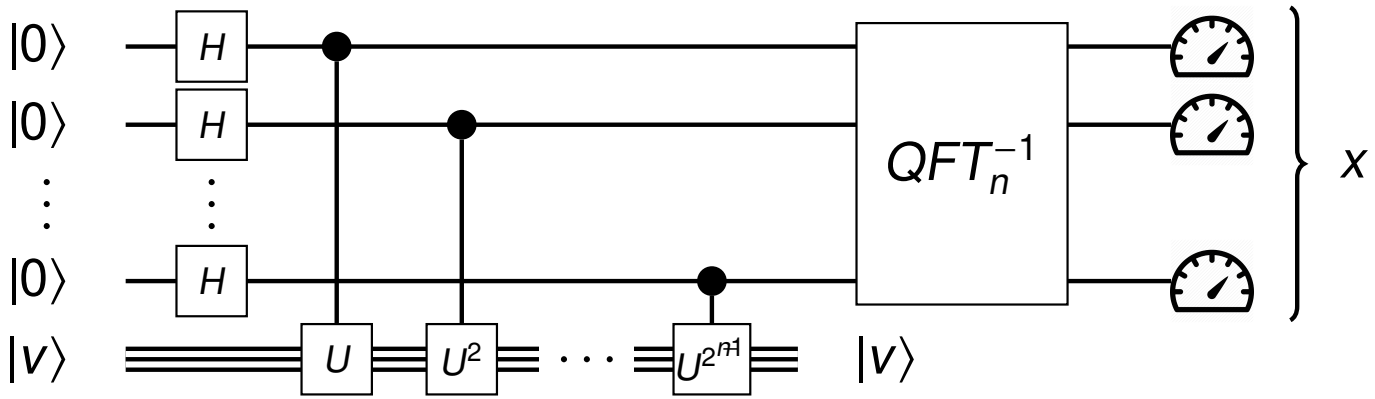
Setting up the stage

- We will compute the bits of the binary decomposition of the phase ϕ one by one
- For simplicity, suppose that ϕ can be expressed exactly by n bits
  - $\phi = \frac{x}{N}$, where x is an integer between 0 and $N = 2^n$, that is

$$x = x_0 + 2x_1 + \ldots + 2^{n-1}x_{n-1}$$

  - If this is not the case, the algorithm still works but with some error
- The algorithm also requires the ability to apply controlled-U^{2^j} efficiently, for any $0 \leq j \leq n-1$.

Circuit for quantum phase estimation



Analysis

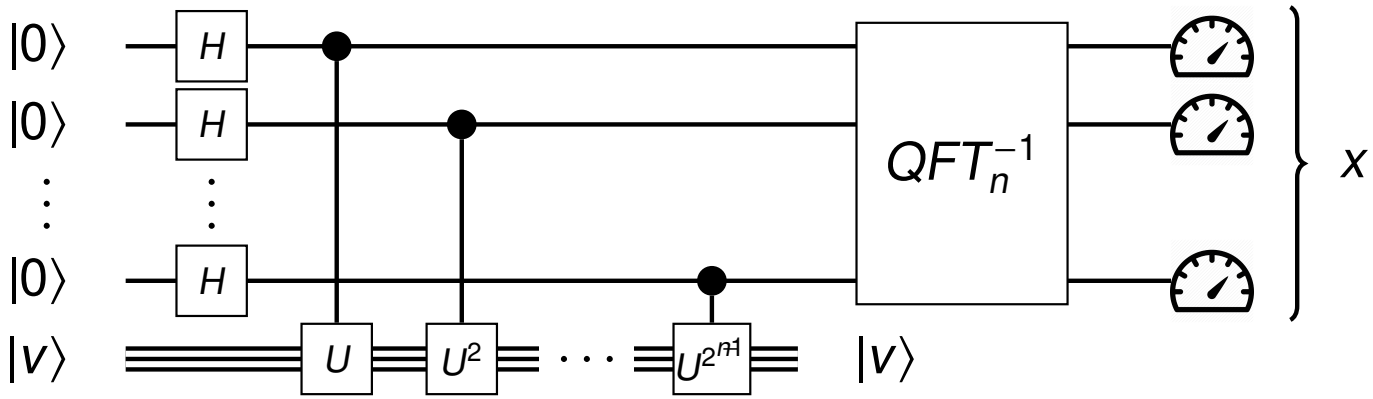
- Let us analyze the state of each qubit just before the QFT
  - Since $U|v\rangle = e^{\frac{2i\pi x}{N}}|v\rangle = \omega^x|v\rangle$, where $\omega = e^{\frac{2i\pi}{N}}$

$$|0\rangle|v\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}\left[|0\rangle + |1\rangle\right]|v\rangle \xrightarrow{\text{c}-U^{2^j}} \frac{1}{\sqrt{2}}\left[|0\rangle + \omega^{2^j x}|1\rangle\right]|v\rangle = \frac{1}{\sqrt{2}}\sum_{y_j\in\{0,1\}} \omega^{y_j 2^j x}|y_j\rangle|v\rangle$$

- Therefore, the resulting state for the n qubits is

$$\frac{1}{\sqrt{2^n}}\sum_{y_0,y_1,\ldots,y_{n-1}} \omega^{\sum_{j=0}^{n-1} y_j 2^j x}|y_0\rangle|y_1\rangle\ldots|y_{n-1}\rangle = \frac{1}{\sqrt{2^n}}\sum_{y\in\mathbb{Z}_N} \omega^{xy}|y\rangle = QFT_n|x\rangle$$

Quantum phase estimation: conclusion



- The state just before QFT_n^{-1} is $QFT_n|x\rangle$
  - which QFT_n^{-1} transforms into $|x\rangle$ itself
- Measuring this state yields x with probability 1
  - and therefore the eigenvalue $e^{\frac{2i\pi x}{N}}$ associated to $|v\rangle$

Notes

- If the phase cannot be exactly expressed as n bits, the measurement yields w.h.p. the n most significant bits of its binary decomposition
- Shor's period finding algorithm is actually equivalent to performing phase estimation on the operator

$$U_c : |x\rangle \mapsto |c \cdot x \mod N\rangle$$

  - whose eigenvalues are $e^{\frac{2i\pi}{a}}$, where a is the order of c.

References

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp.
Quantum Amplitude Amplification and Estimation.
Quantum Computation and Quantum Information: A Millennium Volume, 305:53–74, 2002.
arXiv:quant-ph/0005055.

[EucBC] Euclid.
Elements.
c. 300 BC.

[Gro96] Lov K. Grover.
A fast quantum mechanical algorithm for database search.
In Proc. 28th STOC, pages 212–219, 1996.

[Sho94] Peter W. Shor.
Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.
In Proc. 35th FOCS, pages 124–134, 1994.
arXiv:quant-ph/9508027.