LECTURE 5
GROVER'S SEARCH ALGORITHM AND INTRODUCTION TO THE QUANTUM FOURIER TRANSFORM

INF587 Quantum computer science and applications

Thomas Debris-Alazard

Inria, École Polytechnique

THE OBJECTIVE OF THE DAY

- Grover's algorithm
- Introduction to the Quantum Fourier Transform (**QFT**) but by starting with the classical case!

1. Grover's search algorithm
2. Amplitude amplification
3. Introduction to the discrete Fourier transform
4. Quantum Fourier Transform (QFT) over $\mathbb{Z}/2^n\mathbb{Z}$ (integers modulo 2^n): $\mathsf{QFT}_{\mathbb{Z}/2^n\mathbb{Z}}$

GROVER’S SEARCH ALGORITHM

Given some list L, what is the cost for classically finding a fixed x_0?

$\longrightarrow$ It is, a priori, $\sharp L$!

But is it always the case?

Given some list L, what is the cost for classically finding a fixed x_0?

$\longrightarrow$ It is, a priori, $\sharp L$!

But is it always the case? No!

If the list L has some "structure" it can be helpful:

- Sorted list: time $\log \sharp L$ with a dichotomic search
- Hash table: constant time

Our aim with Grover's algorithm: treating quantumly the case where we are given a list without any structure

Search problem:

- **Input**: a function $f : \{0,1\}^n \longrightarrow \{0,1\}$
- **Goal**: find $\mathbf{x} \in \{0,1\}^n$ be such that $f(\mathbf{x}) = 1$

$\longrightarrow$ Can be viewed as a modelling of a data search in an unstructured database $(x, f(\mathbf{x}))_{\mathbf{x} \in \{0,1\}^n}$ of size 2^n (exponential)

Finding a solution: let $t \stackrel{\text{def}}{=} \sharp\left\{\mathbf{x} \in \{0,1\}^n : f(\mathbf{x}) = 1\right\}$

Let $N = \sharp\{0,1\}^n = 2^n$

- Classically a randomized algorithm would need $\Theta\left(\frac{N}{t}\right)$ queries to f and in time $O\left(\frac{N}{t}\ \mathrm{Cost}(f)\right)$
- Grover can solve this problem with only $O\left(\sqrt{\frac{N}{t}}\right)$ queries to f and in time $O\left(\sqrt{\frac{N}{t}}\ \mathrm{Cost}(f)\right)$

Symmetric cryptography: exhaustive search of the secret key with 128 bits in AES (encryption) requires 2^{128} classical operations

$\longrightarrow$ Quantumly: 2^{64} operations which is reachable...

Consequence:

$\longrightarrow$ All secret keys in symmetric encryption have to be size $\times 2$ (at least...)

Grover offers a generic attack against symmetric encryption schemes, but there are many other ways of taking advantage of quantum computers...

- *Quantum Attacks without Superposition Queries: the Offline Simon's Algorithm.* X. Bonnetain, A. Hosoyamada, M. Naya-Plasencia, Y. Sasaki, A. Schrottenloher:

https://eprint.iacr.org/2019/614.pdf

Lower-bound:

Any algorithm solving the search problem for $f : \{0,1\}^n \longrightarrow \{0,1\}$ with t **solutions** needs to make

$$\Omega\left(\sqrt{\frac{2^n}{t}}\right) \text{ queries to } f$$

$\longrightarrow$ Grover's algorithm is "optimal" (up to constants) in the number of queries to f

A good/bad news:

If the Grover's search problem was solvable in time $\log^c 2^n$: any NP-problem could be solvable (with good probability) in polynomial time with a quantum computer. . .

$\longrightarrow$ There are lower-bounds for the running time of quantum algorithms solving some problems!

- *Lecture notes by Ronald de Wolf's , Chapters* 11.

https://arxiv.org/pdf/1907.09415.pdf

First, with quantum parallelism, we build:

$$|\psi\rangle \overset{\text{def}}{=} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle\,|f(x)\rangle$$

(I) Fundamental idea of Grover algorithm:

Write $|\psi\rangle$ as:

$$|\psi\rangle = \sin\theta\,|\psi_{\text{good}}\rangle + \cos\theta\,|\psi_{\text{bad}}\rangle \quad \text{where} \quad \begin{cases} |\psi_{\text{good}}\rangle = \frac{1}{\sqrt{t}} \sum_{\substack{x \in \{0,1\}^n \\ f(x)=1}} |x\rangle\,|f(x)\rangle \\ |\psi_{\text{bad}}\rangle = \frac{1}{\sqrt{2^n - t}} \sum_{\substack{x \in \{0,1\}^n \\ f(x)=0}} |x\rangle\,|f(x)\rangle \end{cases}$$

with $|\psi_{\text{good}}\rangle$ and $|\psi_{\text{bad}}\rangle$ are quantum states by definition of t (number of solutions)

But what is the value of θ?

First, with quantum parallelism, we build:

$$|\psi\rangle \stackrel{\text{def}}{=} \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \, |f(x)\rangle$$

(I) Fundamental idea of Grover algorithm:

Write $|\psi\rangle$ as:

$$|\psi\rangle = \sin\theta \, |\psi_{\text{good}}\rangle + \cos\theta \, |\psi_{\text{bad}}\rangle \quad \text{where} \quad \begin{cases} |\psi_{\text{good}}\rangle = \frac{1}{\sqrt{t}} \sum_{\substack{x \in \{0,1\}^n \\ f(x)=1}} |x\rangle \, |f(x)\rangle \\ |\psi_{\text{bad}}\rangle = \frac{1}{\sqrt{2^n - t}} \sum_{\substack{x \in \{0,1\}^n \\ f(x)=0}} |x\rangle \, |f(x)\rangle \end{cases}$$

with $|\psi_{\text{good}}\rangle$ and $|\psi_{\text{bad}}\rangle$ are quantum states by definition of t (number of solutions)

But what is the value of θ?

$\longrightarrow$ θ is such that $\frac{\sin\theta}{\sqrt{t}} = \frac{1}{\sqrt{2^n}} \iff \theta = \arcsin\sqrt{\frac{t}{2^n}}$ (we need to know t to know θ)

(II) Fundamental idea of Grover algorithm:

Move θ to $\frac{\pi}{2}$!

THE ANGLE θ?

$$|\psi\rangle = \sin\theta \left|\psi_{\text{good}}\right\rangle + \cos\theta \left|\psi_{\text{bad}}\right\rangle \quad \text{where } \left|\psi_{\text{good}}\right\rangle \text{ uniform superposition of solutions}$$

How is θ when there is few solutions, namely $t \ll 2^n$?

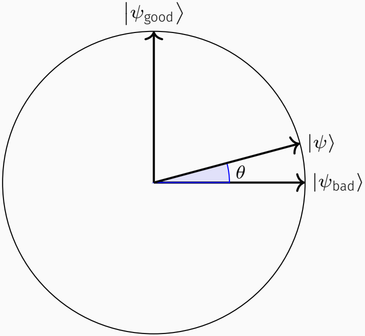
$$|\psi\rangle = \sin\theta \left|\psi_{\text{good}}\right\rangle + \cos\theta \left|\psi_{\text{bad}}\right\rangle \quad \text{where } \left|\psi_{\text{good}}\right\rangle \text{ uniform superposition of solutions}$$

How is θ when there is few solutions, namely $t \ll 2^n$?

$$\longrightarrow \sin\theta = \sqrt{\tfrac{t}{2^n}}, \text{ therefore } \theta \approx \sqrt{\tfrac{t}{2^n}} \approx 0$$

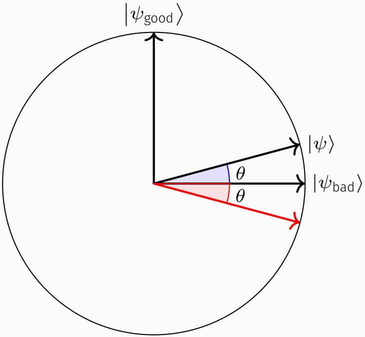
Exercise Session 4: we can make reflexions over a quantum state!

We start by building $|\psi\rangle$



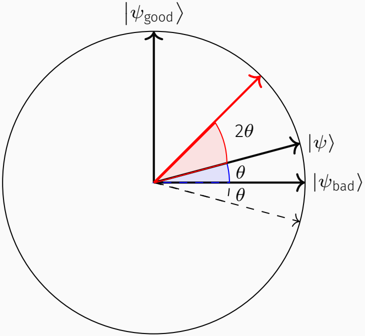
Exercise Session 4: we can make reflexions over a quantum state!

Reflexion over $|\psi_{\text{bad}}\rangle$



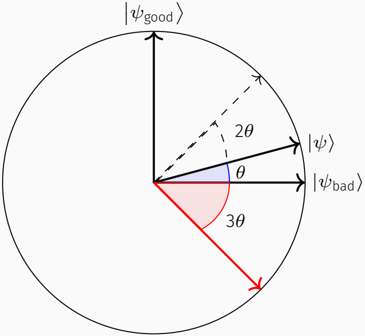
Exercise Session 4: we can make reflexions over a quantum state!

Reflexion over $|\psi\rangle$



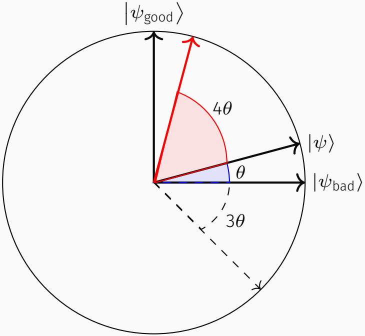
Exercise Session 4: we can make reflexions over a quantum state!

Reflexion over $|\psi_{\text{bad}}\rangle$



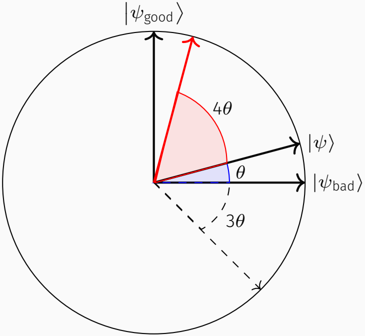
Exercise Session 4: we can make reflexions over a quantum state!

Reflexion over $|\psi\rangle$



Exercise Session 4: we can make reflexions over a quantum state!

and so on up to $\pi/2\ldots$



Number k of iterations to reach $|\psi_{\text{good}}\rangle$: $\theta \to (2k+1)\theta$

Choose the number k of iterations (reflexions over $|\psi_{\text{bad}}\rangle$ and $|\psi\rangle$) such that

$$(2k+1)\theta = \frac{\pi}{2} \iff k = \frac{\pi}{4\theta} - 1 = \frac{\pi}{4\arcsin\sqrt{\frac{t}{2^n}}} - 1 \approx \frac{\pi}{4}\sqrt{\frac{2^n}{t}}$$

$$|\psi_{\text{good}}\rangle = \frac{1}{\sqrt{t}} \sum_{\substack{x \in \{0,1\}^n \\ f(x)=1}} |x\rangle\,|f(x)\rangle \quad \text{and} \quad |\psi_{\text{bad}}\rangle = \frac{1}{\sqrt{2^n - t}} \sum_{\substack{x \in \{0,1\}^n \\ f(x)=0}} |x\rangle\,|f(x)\rangle$$

Reflexion $R_{|\psi_{\text{bad}}\rangle}$ over $|\psi_{\text{bad}}\rangle$:

$$I_n \otimes Z : |x\rangle\,|b\rangle \longmapsto (-1)^b\,|x\rangle\,|b\rangle$$

Reflexion $R_{|\psi\rangle}$ over $|\psi\rangle$:

Exercise session 4: we can build a reflexion $\mathbf{R}_{|\psi\rangle}$ over $|\psi\rangle$ with $O(n)$ elementary gates and two calls to $\mathbf{U}$ which is such that

$$\mathbf{U}\,|0^n\rangle\,|0\rangle = |\psi\rangle \ \left(= \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle\,|f(x)\rangle \right)$$

$\longrightarrow$ Choose $\mathbf{U} = \mathbf{U}_f\left(\mathbf{H}^{\otimes n} \otimes \mathbf{I}_2\right)$

$\longrightarrow$ In Grover's algorithm we crucially used that $|\psi\rangle$ **can be built**!

Proposition:

We have:

$$\cos\alpha\,|\psi_{\text{bad}}\rangle + \sin\alpha\,|\psi_{\text{good}}\rangle \xrightarrow{R_{|\psi\rangle}R_{|\psi_{\text{bad}}\rangle}} \cos(2\theta+\alpha)\,|\psi_{\text{bad}}\rangle + \sin(2\theta+\alpha)\,|\psi_{\text{good}}\rangle$$

Proof:

$$|\psi\rangle = \cos\theta\,|\psi_{\text{bad}}\rangle + \sin\theta\,|\psi_{\text{good}}\rangle \perp \left|\psi^{\perp}\right\rangle = \sin\theta\,|\psi_{\text{bad}}\rangle - \cos\theta\,|\psi_{\text{good}}\rangle$$

From there:

$$|\psi_{\text{bad}}\rangle = \cos\theta\,|\psi\rangle + \sin\theta\left|\psi^{\perp}\right\rangle \quad \text{and} \quad |\psi_{\text{good}}\rangle = \sin\theta\,|\psi\rangle - \cos\theta\left|\psi^{\perp}\right\rangle$$

By definition of the reflexions and trigonometric rules:

$$\begin{aligned}
&R_{|\psi\rangle}R_{|\psi_{\text{bad}}\rangle}\left(\cos\alpha\,|\psi_{\text{bad}}\rangle + \sin\alpha\,|\psi_{\text{good}}\rangle\right) = R_{|\psi\rangle}\left(\cos\alpha\,|\psi_{\text{bad}}\rangle - \sin\alpha\,|\psi_{\text{good}}\rangle\right)\\
&= R_{|\psi\rangle}\left(\cos\alpha\cos\theta - \sin\alpha\sin\theta\right)|\psi\rangle + \left(\cos\alpha\sin\theta + \sin\alpha\cos\theta\right)\left|\psi^{\perp}\right\rangle\\
&= \cos(\alpha+\theta)\,|\psi\rangle - \sin(\alpha+\theta)\left|\psi^{\perp}\right\rangle\\
&= \left(\cos(\alpha+\theta)\cos\theta - \sin\alpha\sin(\theta+\alpha)\right)|\psi_{\text{bad}}\rangle + \left(\cos(\alpha+\theta)\sin\theta + \sin(\alpha+\theta)\cos\theta\right)|\psi_{\text{good}}\rangle\\
&= \cos(2\theta+\alpha)\,|\psi_{\text{bad}}\rangle + \sin(2\theta+\alpha)\,|\psi_{\text{good}}\rangle \qquad \square
\end{aligned}$$

Grover's algorithm:

1. Build $|\psi\rangle = \cos\theta\,|\psi_{\text{bad}}\rangle + \sin\theta\,|\psi_{\text{good}}\rangle$
2. Apply k **times** the unitary $\mathsf{R}_{|\psi\rangle}\mathsf{R}_{|\psi_{\text{bad}}\rangle}$ on the quantum state $|\psi\rangle$
3. Measure, if the last qubit is 1 return the first n qubits; otherwise repeat from step 1

Probability of success (use the previous proposition):

$$P_k = \sin^2(2k\theta + \theta)$$

$\longrightarrow$ How to choose the **number of iterations** k?

Grover's algorithm:

1. Build $|\psi\rangle = \cos\theta\,|\psi_{\text{bad}}\rangle + \sin\theta\,|\psi_{\text{good}}\rangle$
2. Apply k **times** the unitary $\mathsf{R}_{|\psi\rangle}\mathsf{R}_{|\psi_{\text{bad}}\rangle}$ on the quantum state $|\psi\rangle$
3. Measure, if the last qubit is 1 return the first n qubits; otherwise repeat from step 1

Probability of success (use the previous proposition):

$$P_k = \sin^2(2k\theta + \theta)$$

$\longrightarrow$ How to choose the **number of iterations** k?

Choose $k \stackrel{\text{def}}{=} \left\lceil \left(\frac{\pi}{2} - \theta\right) \frac{1}{2\theta} \right\rceil$, then (again some calculations):

$$P_k \geq \frac{1}{4} \quad \text{and } k = O\left(\sqrt{\frac{2^n}{t}}\right) \text{ as } \theta = \arcsin\sqrt{\frac{t}{2^n}}$$

Grover's algorithm finds a solution with constant probability by running $O\left(\sqrt{\frac{2^n}{t}}\right)$ the unitary $\mathsf{R}_{|\psi\rangle}\mathsf{R}_{|\psi_{\text{bad}}\rangle}$.

- $\mathsf{R}_{|\psi_{\text{bad}}\rangle} = \mathsf{I}_n \otimes \mathsf{Z}$: one quantum gate
- $\mathsf{R}_{|\psi\rangle}$: $O(n)$ quantum gates + 2 calls to $\mathsf{U} = \mathsf{U}_f\left(\mathsf{H}^{\otimes n} \otimes \mathsf{I}_2\right)$

Cost of Grover's algorithm:

The cost of Grover's algorithm to find a solution, with constant probability, in the quantum gate model is given by

$$O\left(\sqrt{\frac{2^n}{t}}\,\max(n, T_f)\right)$$

where T_f is the classical running time to compute f.

ISSUES

- Need to run the algorithm $\left\lceil \left(\frac{\pi}{2} - \theta\right) \right\rceil \frac{1}{2\theta}$ where $\theta = \arcsin \sqrt{\frac{t}{2^n}}$ and therefore to know t...

  $\longrightarrow$ If iterations chosen too big, the success probability $\sin((2k+1))^2$ goes down!

- if t is known, can we tweak the algorithm to end up in exactly the good state, namely $P_k = 1$?

$\longrightarrow$ Exercise session to overcome these issues!

AMPLITUDE AMPLIFICATION

$\mathcal{A}$ be a classical/quantum algorithm that can find a solution x (*i.e.*, $f(\mathsf{x}) = 1$) with probability p

$\longrightarrow$ One can repeat $O\left(\frac{1}{p}\right)$ times $\mathcal{A}$ to find a solution with constant probability

Why?

$\mathcal{A}$ be a classical/quantum algorithm that can find a solution $\mathbf{x}$ (*i.e.*, $f(\mathbf{x}) = 1$) with probability p

$\longrightarrow$ One can repeat $O\left(\frac{1}{p}\right)$ times $\mathcal{A}$ to find a solution with constant probability

Why?

Amplitude Amplification:

Assume you have a classical or quantum algorithm $\mathcal{A}$ (**without measurement**) that can find a solution $\mathbf{x}$ to the search problem ($f(\mathbf{x}) = 1$) in time T with probability p.

If f is computable in time T_f, then we can compute (quantumly) a solution in time $O\left(\frac{T}{\sqrt{p}}\max(n, T_f)\right)$ with success probability $\geq C$ (constant).

Pick a random $\mathbf{x} \in \{0,1\}^n$ and output $\mathbf{x}$

$\longrightarrow$ This algorithm runs in time $O(n)$ and it finds a solution with probability $p = \frac{t}{2^n}$

Using amplitude amplification: you can find a solution in time $\approx \sqrt{\frac{2^n}{t}}$

Grover: quantization of the random search in an unstructured data set...

Amplitude amplification is more useful when we know algorithms better than random search

$\longrightarrow$ It also gives a quadratic speed-up for these algorithms!

Lecture 4:

If $\mathcal{A}$ is quantum: measurements only at the end of the computation and starts from $\left|0^m\right\rangle$

$\longrightarrow$ Before the final measurement: $\mathcal{A}$ outputs a state $|\psi\rangle$, and measuring the output register gives a solution $\mathbf{x}$ with probability p

$$\mathcal{A}\left|0^m\right\rangle = |\psi\rangle = \sum_{\mathrm{x}\in\{0,1\}^n} \alpha_{\mathrm{x}} |\mathrm{x}\rangle |\varphi_{\mathrm{x}}\rangle\,, \text{ where } \sum_{\mathrm{x}:f(\mathrm{x})=1} |\alpha_{\mathrm{x}}|^2 = p$$

Write:
$$|\psi\rangle = \sin\theta\left|\psi_{\text{good}}\right\rangle + \cos\theta\left|\psi_{\text{bad}}\right\rangle \quad \text{where } \left|\psi_{\text{good}}\right\rangle \stackrel{\text{def}}{=} \frac{1}{\sin\theta}\sum_{\substack{\mathrm{x}\in\{0,1\}^n \\ f(\mathrm{x})=1}} \alpha_{\mathrm{x}} |\mathrm{x}\rangle |\alpha_{\mathrm{x}}\rangle$$

$$\text{where } \sin\theta = \sqrt{p}$$

Lecture 4:

If $\mathcal{A}$ is quantum: measurements only at the end of the computation and starts from $\left|0^m\right\rangle$

$\longrightarrow$ Before the final measurement: $\mathcal{A}$ outputs a state $|\psi\rangle$, and measuring the output register gives a solution $\mathbf{x}$ with probability p

$$\mathcal{A}\left|0^m\right\rangle = |\psi\rangle = \sum_{\mathsf{x}\in\{0,1\}^n} \alpha_{\mathsf{x}} |\mathsf{x}\rangle |\varphi_{\mathsf{x}}\rangle \text{ , where } \sum_{\mathsf{x}:f(\mathsf{x})=1} |\alpha_{\mathsf{x}}|^2 = p$$

Write:

$$|\psi\rangle = \sin\theta \left|\psi_{\text{good}}\right\rangle + \cos\theta \left|\psi_{\text{bad}}\right\rangle \quad \text{where } \left|\psi_{\text{good}}\right\rangle \stackrel{\text{def}}{=} \frac{1}{\sin\theta} \sum_{\substack{\mathsf{x}\in\{0,1\}^n \\ f(\mathsf{x})=1}} \alpha_{\mathsf{x}} |\mathsf{x}\rangle |\alpha_{\mathsf{x}}\rangle$$

$$\text{where } \sin\theta = \sqrt{p}$$

Run Grover's algorithm with the reflexions $\mathsf{R}_{|\psi_{\text{bad}}\rangle} : |\mathsf{x}\rangle|\mathsf{y}\rangle \mapsto (-1)^{f(\mathsf{x})}|\mathsf{x}\rangle|\mathsf{y}\rangle$ (see Exercise Session to compute this unitary) and $\mathsf{R}_{|\psi\rangle}$ over $|\psi\rangle$ but:

$\mathsf{R}_{|\psi\rangle} \neq O(n)$ quantum gates + 2 calls to $\mathsf{U} = \mathsf{U}_f\left(\mathsf{H}^n \otimes \mathsf{I}_2\right)$ which was designed to build

$$\frac{1}{\sqrt{2^n}} \sum_{\mathsf{x}} |\mathsf{x}\rangle |f(\mathsf{x})\rangle \ldots$$

Amplitude amplification: $\mathsf{R}_{|\psi\rangle}$ is $O(n)$ quantum gates + 1 call to $\mathsf{U} = \mathcal{A}$ and 1 call to $\mathsf{U}^{-1} = \mathcal{A}^{-1}$

When performing amplitude amplification on a quantum algorithm $\mathcal{A}$, we supposed it performs no measurements (at least we restrict $\mathcal{A}$ before its final measurement)

$\longrightarrow$ To be able to perform $\mathcal{A}^{-1}$

Grover's search algorithm in amplitude amplification shows a strong statement. Given

$$|\psi\rangle = \alpha |\psi_V\rangle + \beta \left|\psi_V^\perp\right\rangle \text{ where } |\psi_V\rangle \in \mathrm{Span}\left(|x\rangle : f(x) = 1\right) \text{ and } \left|\psi_V^\perp\right\rangle \in \mathrm{Span}\left(|x\rangle : f(x) = 1\right)^\perp$$

After amplitude amplification: $|\psi'\rangle \approx |\psi_V\rangle$

(even equal with exact grover when amplitude α is known)

Be careful:

To run amplitude amplification: you need to be able to build $|\psi\rangle$ ...

Lecture 4: given a deterministic $\mathcal{A}$, one can run $\mathsf{U}_{\mathcal{A}}$ in $\approx$ same time

If $\mathcal{A}$ is randomized?

Classical modelization (think R be the seed of a pseudo-random generator):

$$\mathcal{A} : \text{pick a random } \mathsf{R} \in \{0,1\}^r, \text{ compute } \mathcal{A}(\mathsf{R}) \text{ to get some outcome } \mathsf{x}_{\mathsf{R}}$$

$\longrightarrow$ Randomness chosen at the beginning: the algorithm can be interpreted as deterministic

APPLICATION: HOW DO WE QUANTUMLY COMPUTE RANDOMIZED ALGORITHMS?

Lecture 4: given a deterministic $\mathcal{A}$, one can run $\mathsf{U}_{\mathcal{A}}$ in $\approx$ same time

If $\mathcal{A}$ is randomized?

Classical modelization (think R be the seed of a pseudo-random generator):

$$\mathcal{A} : \text{pick a random } \mathsf{R} \in \{0,1\}^r, \text{ compute } \mathcal{A}(\mathsf{R}) \text{ to get some outcome } \mathsf{x}_\mathsf{R}$$

$\longrightarrow$ Randomness chosen at the beginning: the algorithm can be interpreted as **deterministic**

$$\mathsf{U}_{\mathcal{A}}(|\mathsf{R}\rangle\,|\mathsf{y}\rangle) = |\mathsf{R}\rangle\,|\mathsf{y} + \mathsf{x}_\mathsf{R}\rangle$$

$$|0^r\rangle\,|0^n\rangle \xrightarrow{\mathsf{H}^{\otimes r}\otimes \mathsf{I}_r} \frac{1}{\sqrt{2^r}} \sum_{\mathsf{R}\in\{0,1\}^r} |\mathsf{R}\rangle\,|0^n\rangle \xrightarrow{\mathsf{U}_{\mathcal{A}}} \frac{1}{\sqrt{2^r}} \sum_{\mathsf{R}\in\{0,1\}^r} |\mathsf{R}\rangle\,|\mathsf{x}_\mathsf{R}\rangle$$

measuring outputs a solution with probability p

$\longrightarrow$ We can use amplitude amplification on this algorithm!

DISCRETE FOURIER TRANSFORM

- $(G, +)$ be a finite Abelian group
- Character group: $\widehat{G} = \{\chi_g : g \in G\} \cong G$, set of characters = homomorphism from G to the unit complex circle $\mathbb{U} = \{z \in \mathbb{C} : |z| = 1\}$

$$\begin{aligned} \chi_g : G &\longrightarrow \mathbb{U} \\ x &\longmapsto \chi_g(x), \text{ such that} \end{aligned}$$

$$\forall x, y \in G,\ \chi_g(x+y) = \chi_g(x)\chi_g(y)$$

Examples:

- $G = \mathbb{F}_2^n = \underbrace{\mathbb{F}_2 \times \cdots \times \mathbb{F}_2}_{n \text{ times}}$,[a]

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n, \quad \chi_{\mathbf{x}}(\mathbf{y}) = (-1)^{\mathbf{x}\cdot\mathbf{y}} \quad \text{where } \mathbf{x}\cdot\mathbf{y} = \sum_{i=1}^{n} x_i y_i$$

- $G = \mathbb{Z}/2^n\mathbb{Z}$,

$$\forall x, y \in \mathbb{Z}/2^n\mathbb{Z}, \quad \chi_x(y) = \mathrm{e}^{-\frac{2i\pi xy}{2^n}}$$

[a] $\mathbb{F}_2$ binary field, $\{0, 1\}$ embedded with $\oplus$

Nice reading about characters on finite Abelian groups:

https://kconrad.math.uconn.edu/blurbs/grouptheory/charthy.pdf

$$\sum_{g\in G}\chi_x(g)\overline{\chi_y}(g) = \begin{cases} \sharp G & \text{if } \chi_x = \chi_y \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \sum_{\chi\in\widehat{G}}\chi(x)\overline{\chi}(y) = \begin{cases} \sharp G & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

- The matrix $\left(\frac{\chi_x(y)}{\sqrt{\sharp G}}\right)_{x,y\in G}$ is unitary, in particular:

$$\left(\frac{\chi_x}{\sqrt{\sharp G}}\right)_{x\in G} \text{ orthonormal basis for the scalar product } \langle f, g\rangle = \sum_{y\in G} f(y)\overline{g}(y)$$

$\left(\frac{\chi_x}{\sqrt{\sharp G}}\right)_{x\in G}$ sometimes called the "Fourier basis"

- The translation operator is diagonal in the Fourier basis

$$\begin{array}{rcl} \tau_a : (G \to \mathbb{C}) & \longrightarrow & (G \to \mathbb{C}) \\ f & \longmapsto & \tau_a(f) : x \in G \mapsto f(x+a) \text{ then} \end{array}$$

$$\tau_x(\chi_y) = \underbrace{\chi_y(a)}_{\text{egeinvalue}} \underbrace{\chi_y}_{\text{eigenvector}}$$

Exercise:

1. Prove that for any character $\chi \in \widehat{G}$,

$$\sum_{g \in G} \chi(g) = \begin{cases} \sharp G & \text{if } \chi = 1 \\ 0 & \text{otherwise} \end{cases}$$

2. How do you deduce from that

$$\sum_{g \in G} \chi_x(g)\overline{\chi_y}(g) = \begin{cases} \sharp G & \text{if } \chi_x = \chi_y \\ 0 & \text{otherwise} \end{cases}$$

3. Consider the function f_g

$$\begin{aligned} f_g : \widehat{G} &\longrightarrow G \\ \chi &\longmapsto \chi(g), \text{ such that} \end{aligned}$$

   What can you say about f_g?

4. How can you deduce from the previous point that we also have

$$\sum_{\chi \in \widehat{G}} \chi(x)\overline{\chi}(y) = \begin{cases} \sharp G & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

Orthogonal subgroup:

For a subgroup H of G we denote by $H^\perp$ the orthogonal subgroup defined by

$$H^\perp \stackrel{\text{def}}{=} \{g \in G \ : \ \forall h \in H,\ \chi_g(h) = 1\}$$

$\longrightarrow$ Important concept in Simon/Shor's algorithm! (see Lecture 6)

$$\sum_{h \in H} \chi_g(h) = \begin{cases} \sharp H & \text{if } g \in H^\perp \\ 0 & \text{otherwise} \end{cases}$$

Fourier transform:

Given a finite abelian group G and $f : G \longrightarrow \mathbb{C}$, its Fourier transform is

$$\forall x \in G, \quad \widehat{f}(x) = \frac{1}{\sqrt{\sharp G}} \sum_{y \in G} f(y) \overline{\chi_x}(y)$$

Notice that:

$$\widehat{f}(x) = \left\langle f, \frac{\chi_x}{\sqrt{\sharp G}} \right\rangle \text{ where standard scalar product } \langle \cdot, \cdot \rangle \text{ over functions } G \longrightarrow \mathbb{C}$$

$\left(\frac{\chi_x}{\sqrt{\sharp G}} \right)_{x \in G}$ orthonormal basis for this scalar product and $\widehat{f}(x)$: x-thm coefficient of f in this basis.

Exercise:

Compute the Fourier transform of the following function $\mathbb{F}_2^n \longrightarrow \mathbb{C}$,

- $f(\mathbf{0}) = 1$ and 0 otherwise
- $\forall \mathbf{x} \in \mathbb{F}_2^n, f(\mathbf{x}) = \frac{1}{2^n}$
- Does it remind you something?

Classical Fourier Transform	Quantum Fourier Transform: QFT_G
$f = (f(x))_{x\in G}$	$\lvert\psi_f\rangle = \sum_{x\in G} f(x)\lvert x\rangle$ $(\lVert f\rVert_2 = 1)$
$\widehat{f}(x) = \frac{1}{\sqrt{\sharp G}} \sum_{y\in G} f(y)\overline{\chi_x}(y)$	$\mathrm{QFT}_G \lvert\psi\rangle \overset{\text{def}}{=} \widehat{\lvert\psi_f\rangle} = \sum_{x\in G} \widehat{f}(x)\lvert x\rangle$

$\longrightarrow$ In particular: $\forall x \in G$, $\mathrm{QFT}_G \lvert x\rangle = \frac{1}{\sqrt{\sharp G}} \sum_{y\in G} \overline{\chi_y}(x)\lvert y\rangle$

$\Big($It corresponds to the fact that $\widehat{\delta_x}(y) = \frac{1}{\sqrt{\sharp G}}\overline{\chi_y}(x)$ where δ_x Kronecker symbol and δ_x"="$\lvert x\rangle$ $\Big)$

Exercise:

Show that $\lvert\psi_f\rangle$ is a quantum state

Formally, given any finite group G: $(\lvert x\rangle)_{x\in G}$ denotes an orthonormal basis of an Hilbert space of dimension $\sharp G$

Given **x**, what is the cost for (classically) computing $\widehat{f}(x)$?

Given **x**, what is the cost for (classically) computing $\widehat{f}(x)$?

$\longrightarrow$ It costs $\sharp G \ldots$ Be careful: in practice $\sharp G = 2^n$

What is the cost for (classically) computing $\widehat{f}$, namely all the $\widehat{f}(x)$'s?

Given **x**, what is the cost for (classically) computing $\widehat{f}(x)$?

$\longrightarrow$ It costs $\sharp G \ldots$ Be careful: in practice $\sharp G = 2^n$

What is the cost for (classically) computing $\widehat{f}$, namely all the $\widehat{f}(x)$'s?

$\longrightarrow$ It costs naively $(\sharp G)^2$

$\longrightarrow$ We can do much better to compute $\widehat{f}$

The Fast Fourier Transform (FFT): computing $\widehat{f}$ costs $O(\sharp G \log \sharp G)$ (in most cases. . .)

Suppose that $G = \mathbb{Z}/2^n\mathbb{Z}$, in particular $\sharp G = 2^n$

$$N \stackrel{\text{def}}{=} 2^n \text{ and } \omega_N \stackrel{\text{def}}{=} \mathrm{e}^{-\frac{2i\pi}{N}}$$

Divide and Conquer strategy:

$$\begin{aligned}
\widehat{f}(j) &= \frac{1}{\sqrt{2^n}} \sum_{k=0}^{N-1} f(k)\omega_N^{-jk} \\
&= \frac{1}{\sqrt{N}} \left(\sum_{k \text{ even}} f(k)\omega_N^{-jk} + \omega_N^{-j} \sum_{k \text{ odd}} f(k)\omega_N^{-j(k-1)} \right) \\
&= \frac{1}{\sqrt{2}} \left(\frac{1}{\sqrt{N/2}} \sum_{k \text{ even}} f(k)\omega_{N/2}^{-j/2k} + \omega_N^{-j} \frac{1}{\sqrt{N/2}} \sum_{k \text{ odd}} f(k)\omega_{N/2}^{-j/2(k-1)} \right)
\end{aligned}$$

$\longrightarrow$ Therefore we reduce the computation of $\widehat{f}(j)$ to two Fourier transforms over $\mathbb{Z}/2^{n-1}\mathbb{Z}$

Cost: $T(2^n) = 2T(2^{n-1}) + O(2^n)$, therefore $T(2^n) = O(2^n \underbrace{\log(2^n)}_{\text{rec. calls}}) = O\left(n2^n\right)$

Computing the quantum Fourier transform:

- QFT_G can be implemented by a quantum circuit of size $O\left(\log^3 \sharp G\right)$ for any arbitrary finite Abelian group G
- $\mathrm{QFT}_{\mathbb{Z}/N\mathbb{Z}}$ can be implemented by a quantum circuit of size $O\left(\log^3 N\right)$
- $\mathrm{QFT}_{\mathbb{Z}/2^n\mathbb{Z}}$ can be implemented by a quantum circuit of size $O\left(n^2\right)$ (here $n = \log 2^n = \log \sharp\mathbb{Z}/2^n\mathbb{Z}$)
- $\mathrm{QFT}_{\mathbb{Z}/2^n\mathbb{Z}}$ can be implemented up to some accuracy [a] by a quantum circuit of size $O\left(n \log n\right)$

[a] for the norm operator

$\longrightarrow$ Exponentially faster than computing the classical Fourier transform, even with the FFT trick which is for instance $O(n2^n)$ in the case of $\mathbb{Z}/2^n\mathbb{Z}$

Quantum Fourier over $\mathbb{F}_2^n$ (the set $\{0,1\}^n$ with the $\oplus$ operation term by term)?

$\longrightarrow$ Characters are given by $\chi_{\mathbf{x}}(\mathbf{y}) = (-1)^{\mathbf{x}\cdot\mathbf{y}}$ where $\mathbf{x}\cdot\mathbf{y} = \sum_{i=1}^{n} x_i y_i$

$$\widehat{f}(\mathbf{x}) = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{y}\in\mathbb{F}_2^n} (-1)^{\mathbf{x}\cdot\mathbf{y}} f(\mathbf{y})$$

Quantum Fourier Transform in $\mathbb{F}_2^n$:

$$\mathrm{QFT}_{\mathbb{F}_2^n} |\mathbf{x}\rangle = \frac{1}{\sqrt{2^n}} \sum_{\mathbf{y}\in\mathbb{F}_2^n} (-1)^{\mathbf{x}\cdot\mathbf{y}} |\mathbf{y}\rangle$$

$\longrightarrow$ $\mathrm{QFT}_{\mathbb{F}_2^n} = \mathrm{H}^{\otimes n}$ and its cost: $O(n)$

QUANTUM FOURIER TRANSFORM $\mathrm{QFT}_{\mathbb{Z}/2^n\mathbb{Z}}$

Give an efficient quantum circuit for computing $\mathbf{QFT}_{\mathbb{Z}/2^n\mathbb{Z}}$

Gates that we will use:

$$\mathbf{H} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \text{ (Hadamard)} \qquad \mathbf{R}_s = \begin{pmatrix} 1 & 0 \\ 0 & \mathrm{e}^{\frac{2i\pi}{2^s}} \end{pmatrix} \text{ (Phase rotation)}$$

$$\mathbf{C}\text{-}\mathbf{R}_s : \begin{cases} |0\rangle\,|x\rangle \mapsto |0\rangle\,|x\rangle \\ |1\rangle\,|x\rangle \mapsto |1\rangle\,\mathbf{R}_s\,|x\rangle \end{cases} \qquad \text{(Controlled-}\mathbf{R}_s\text{)}$$

Notation:

For any integer $j \in [\![0, 2^n - 1]\!]$, binary decomposition $j = j_1 \dots j_n$ where j_1 bit of highest weight

$$j = \sum_{\ell=1}^{n} 2^{n-\ell} j_\ell$$

For any $x \in [\![0, 2^n - 1]\!]$,

$$|x\rangle = |x_1, \dots, x_n\rangle$$

$$\begin{aligned}
\mathrm{QFT}_{\mathbb{Z}/2^n\mathbb{Z}} |k\rangle &= \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \mathrm{e}^{\frac{2ikj\pi}{2^n}} |j\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \mathrm{e}^{2i\pi k \left(\sum_{\ell=1}^{n} 2^{-\ell} j_\ell\right)} |j_1, \dots, j_n\rangle \\
&= \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} \prod_{\ell=1}^{n} \mathrm{e}^{2i\pi k 2^{-\ell} j_\ell} |j_1, \dots, j_n\rangle \\
&= \bigotimes_{\ell=1}^{n} \left(\frac{|0\rangle + \mathrm{e}^{2i\pi k 2^{-\ell}} |1\rangle}{\sqrt{2}} \right)
\end{aligned}$$

$\longrightarrow$ $\mathrm{QFT}_{\mathbb{Z}/2^n\mathbb{Z}} |k\rangle$ is a separable quantum state!

Be careful: we crucially use that we work in $\mathbb{Z}/2^n\mathbb{Z}$

How to compute $\bigotimes_{\ell=1}^{n}\left(\frac{|0\rangle+e^{2i\pi k2^{-\ell}}|1\rangle}{\sqrt{2}}\right)$?

Idea: write the binary decomposition of $k2^{-\ell}$

$$
\begin{aligned}
e^{2i\pi k2^{-\ell}} &= e^{2i\pi\left(\sum_{m=1}^{n} 2^{n-m-\ell}k_m\right)} \\
&= e^{2i\pi\left(\sum_{m=n-\ell+1}^{n} 2^{n-m-\ell}k_m\right)} \qquad (\text{if } m \leq n-\ell, \text{ then } 2^{n-m-\ell} \in \mathbb{N}) \\
&= e^{2i\pi\left(\sum_{m=1}^{\ell} 2^{-m}k_{n-\ell+m}\right)} \qquad (n-\ell-m_{\text{old}} \longleftrightarrow -m_{\text{new}})
\end{aligned}
$$

$$
\frac{|0\rangle+e^{2i\pi k2^{-\ell}}|1\rangle}{\sqrt{2}} = \frac{|0\rangle+e^{2i\pi 0.k_{n-\ell+1}\cdots k_n}|1\rangle}{\sqrt{2}}
$$

where for any integer $j = j_1 \dots j_p$

$$
0.j_1 \dots j_p \overset{\text{def}}{=} \frac{j}{2^p} = \sum_{\ell=1}^{p} 2^{-\ell} j_\ell
$$

$\mathrm{QFT}_{\mathbb{Z}/2^n\mathbb{Z}} |k\rangle$ is equal to:

$$\left(\frac{|0\rangle + \mathrm{e}^{2i\pi 0.k_n} |1\rangle}{\sqrt{2}}\right) \bigotimes \left(\frac{|0\rangle + \mathrm{e}^{2i\pi 0.k_{n-1}k_n} |1\rangle}{\sqrt{2}}\right) \bigotimes \cdots \bigotimes \left(\frac{|0\rangle + \mathrm{e}^{2i\pi 0.k_1 k_{n-\ell}\cdots k_n} |1\rangle}{\sqrt{2}}\right)$$

where

$$k = \sum_{\ell=1}^{n} 2^{n-\ell} k_\ell \in [\![0, 2^n - 1]\!] \quad \text{and} \quad 0.k_n \ldots k_{n+1-p} = \sum_{\ell=1}^{p} 2^{-\ell} k_{n+1-\ell} \in [0,1)$$

To build this quantum state, we will crucially use:

$$\text{C-R}_s |b\rangle |1\rangle = |b\rangle \, \mathrm{e}^{\frac{2i\pi b}{2^s}} |1\rangle = |b\rangle \, \mathrm{e}^{2i\pi 0.0^{s-1}b} |1\rangle \quad \text{where} \;\; 0.0^{s-1}b = 0.\underbrace{0\ldots 0}_{s \text{ times}}b$$

$$\text{C-R}_s |b\rangle |0\rangle = |b\rangle |0\rangle$$

Aim: starting from $|k_1, k_2, k_3\rangle$ building

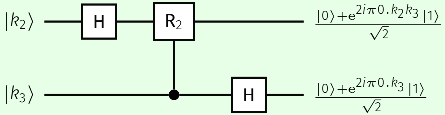
$$\left(\frac{|0\rangle + e^{2i\pi 0.k_3}|1\rangle}{\sqrt{2}}\right) \bigotimes \left(\frac{|0\rangle + e^{2i\pi 0.k_2k_3}|1\rangle}{\sqrt{2}}\right) \bigotimes \left(\frac{|0\rangle + e^{2i\pi 0.k_1k_2k_3}|1\rangle}{\sqrt{2}}\right)$$

1. Sending $|k_3\rangle$ through **H**:

$$|k_3\rangle \xrightarrow{\mathrm{H}} \frac{|0\rangle + (-1)^{k_3}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{2i\pi 0.k_3}|1\rangle}{\sqrt{2}} \qquad \left(0.k_3 = 0 \text{ if } k_3 = 0 \text{ or } \tfrac{1}{2} \text{ if } k_3 = 1\right)$$

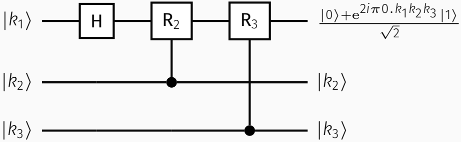
1. Sending $|k_3\rangle\,|k_2\rangle$ through $\mathrm{I}_2 \otimes$ **H** and then **C-R**$_2$:

$$|k_3\rangle\,|k_2\rangle \xrightarrow{\mathrm{I}_2\otimes\mathrm{H}} |k_3\rangle \frac{|0\rangle + e^{2i\pi 0.k_2}|1\rangle}{\sqrt{2}} \xrightarrow{\mathrm{C\text{-}R}_2} |k_3\rangle \frac{|0\rangle + e^{2i\pi 0.0k_3} e^{2i\pi 0.k_2}|1\rangle}{\sqrt{2}} = |k_3\rangle \frac{|0\rangle + e^{2i\pi 0.k_2k_3}|1\rangle}{\sqrt{2}}$$



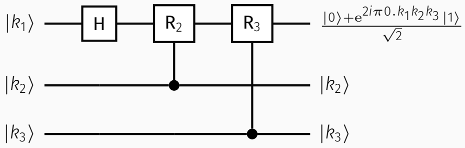
THE CASE 2^3

3. Sending $|k_3\rangle\,|k_2\rangle\,|k_1\rangle$ through the following circuit:



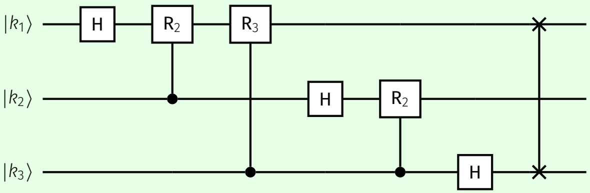
Combining this with the previous circuit gives almost the good state not in the good order: swap!

3. Sending $|k_3\rangle\,|k_2\rangle\,|k_1\rangle$ through the following circuit:



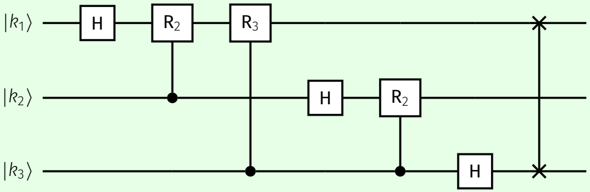
Combining this with the previous circuit gives almost the good state not in the good order: swap!

QFT over $\mathbb{Z}/8\mathbb{Z}$:



GENERAL CASE

QFT over $\mathbb{Z}/8\mathbb{Z}$:



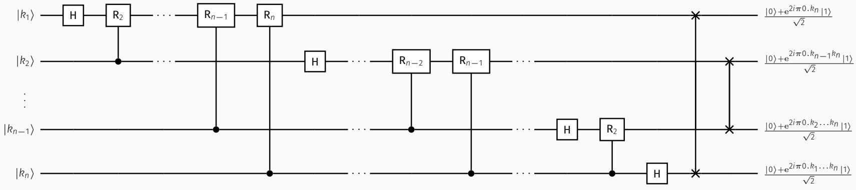
The general case $\mathbb{Z}/2^n\mathbb{Z}$ will follow the same pattern: $O(n^2) + \textbf{SWAP} = O(n^2) = O\left(\log 2^n\right)^2$ gates

$\longrightarrow$ In particular gates $R_2, \ldots, R_n$ are used!

But $\mathbf{R}_s = \begin{pmatrix} 1 & 0 \\ 0 & e^{\frac{2i\pi}{2^s}} \end{pmatrix}$ is very close to the identity if $s \gg \log n$

If one allows errors: removing all the $\mathbf{R}_s$ for $s \geq C \log n$ (with C constant) will lead to the result with accuracy $\leq \frac{1}{n}$

$\longrightarrow$ In that case: only $O(n \log n)$ gates!


$|k_1\rangle$
$|k_2\rangle$
$|k_{n-1}\rangle$
$|k_n\rangle$
H
R_2
R_{n-1}
R_n
R_{n-2}
$\frac{|0\rangle+e^{2i\pi 0.k_n}|1\rangle}{\sqrt{2}}$
$\frac{|0\rangle+e^{2i\pi 0.k_{n-1}k_n}|1\rangle}{\sqrt{2}}$
$\frac{|0\rangle+e^{2i\pi 0.k_2\ldots k_n}|1\rangle}{\sqrt{2}}$
$\frac{|0\rangle+e^{2i\pi 0.k_1\ldots k_n}|1\rangle}{\sqrt{2}}$

EXERCISE SESSION