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

Quantum Information and Computing

Thomas Debris-Alazard

Inria, École Polytechnique

THE OBJECTIVE OF THE DAY

- Grover's algorithm
- Introduction to the Quantum Fourier Transform $\left(\text{QFT}\right)$ 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 $\left(\mathsf{QFT}\right)$ over $\mathbb{Z}/2^n\mathbb{Z}$ $\left(\text{integers modulo } 2^n\right)$: $\mathsf{QFT}_{\mathbb{Z}/2^n\mathbb{Z}}$

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 binary search
- Hash table: constant time $\big($in the average/amortized complexity model$\big)$

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 $\mathbf{x} \in \{0,1\}^n$ such that $f(\mathbf{x}) = 1$

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

of size 2^n $\big($exponential$\big)$

Finding a solution:

$$\text{Let } N \stackrel{\text{def}}{=} \sharp\{0,1\}^n = 2^n \quad \text{and} \quad t \stackrel{\text{def}}{=} \sharp\left\{\mathbf{x} \in \{0,1\}^n : f(\mathbf{x}) = 1\right\}$$

- 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 for the secret key with 128 bits in AES $\left(\text{encryption}\right)$ 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 $\times 2$ $\left(\text{at least...}\right)$

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

- *Breaking Symmetric Cryptosystems using Quantum Period Finding.* M. Kaplan, G. Leurent, A. Leverrier, M. Naya-Plasencia

`https://arxiv.org/pdf/1602.05973`

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"** $\big($up to constants$\big)$ in the number of queries to f

A good/bad news:

If Grover's search problem was solvable in time $\log^c 2^n = n^c$: any NP-problem could be solvable $\big($with good probability$\big)$ 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*, *Chapter* 11

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

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's algorithm:

Write $|\psi\rangle$ as:

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

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

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's algorithm:

Write $|\psi\rangle$ as:

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

with $\left|\psi_{\text{good}}\right\rangle$ 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's algorithm:

Move θ to $\frac{\pi}{2}$!

THE ANGLE θ?

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

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

THE ANGLE θ?

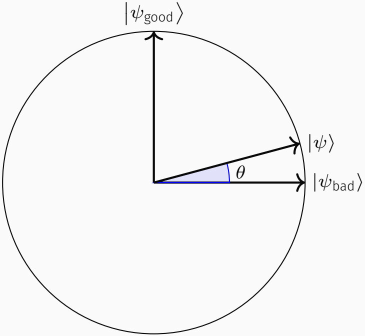
$$|\psi\rangle = \sin\theta \left|\psi_{\text{good}}\right\rangle + \cos\theta \left|\psi_{\text{bad}}\right\rangle \quad \text{where } \left|\psi_{\text{good}}\right\rangle \text{ uniform superposition of solutions}$$

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

$$\longrightarrow \sin\theta = \sqrt{\tfrac{t}{2^n}}, \quad \text{therefore} \quad \theta \approx \sqrt{\tfrac{t}{2^n}} \approx 0 \quad \text{and} \quad |\psi\rangle \approx |\psi_{\text{bad}}\rangle$$

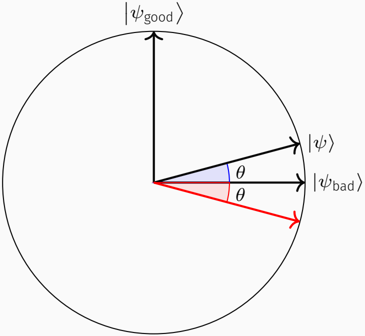
Exercise Session 4: we can make reflections over a quantum state!

We start by building $|\psi\rangle$



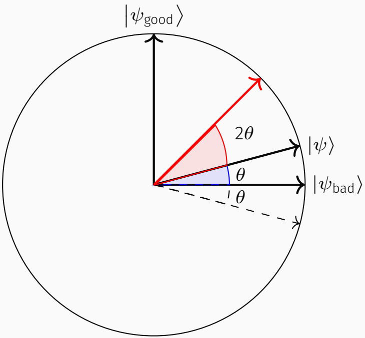
Exercise Session 4: we can make reflections over a quantum state!

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



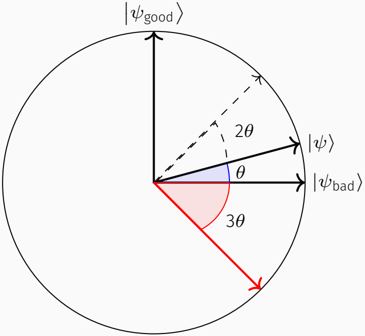
Exercise Session 4: we can make reflections over a quantum state!

Reflection over $|\psi\rangle$



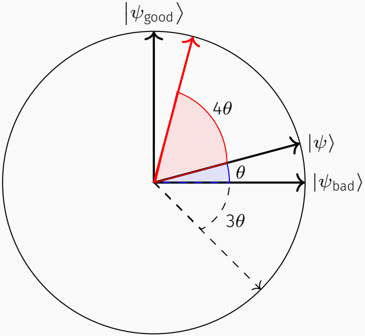
Exercise Session 4: we can make reflections over a quantum state!

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



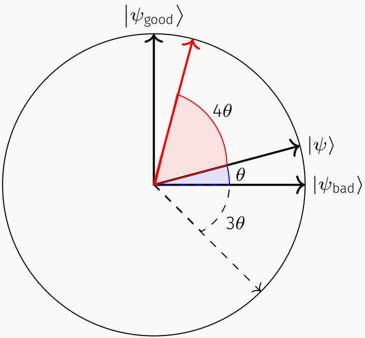
Exercise Session 4: we can make reflections over a quantum state!

Reflection over $|\psi\rangle$



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

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



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

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

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

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

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

$$\text{Id} \otimes Z : |x\rangle\, |b\rangle \longmapsto (-1)^b\, |x\rangle\, |b\rangle$$

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

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

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

$$\longrightarrow \text{Choose } \mathbf{U} = \mathbf{U}_f \cdot \left(\mathbf{H}^{\otimes n} \otimes \mathbf{Id}\right)$$

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

Proposition:

We have:

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

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 reflections and trigonometric rules:

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

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 $\Big($use the previous proposition$\Big)$:

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

How to choose the **number of iterations** k?

Grover's algorithm:

1. Build $|\psi\rangle = \cos\theta\,|\psi_{\mathrm{bad}}\rangle + \sin\theta\,|\psi_{\mathrm{good}}\rangle$
2. Apply k **times** the unitary $\mathsf{R}_{|\psi\rangle}\mathsf{R}_{|\psi_{\mathrm{bad}}\rangle}$ 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 $\Big($use the previous proposition$\Big)$:

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

How to choose the **number of iterations** k?

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

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

Grover's algorithm finds a solution with constant probability

$\left(\text{bounded away from 0 by a constant}\right)$

by running the unitary $\mathsf{R}_{|\psi\rangle}\mathsf{R}_{|\psi_{\text{bad}}\rangle}$ a $O\left(\sqrt{\frac{2^n}{t}}\right)$ number of times

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

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 $\theta = \textbf{arcsin} \sqrt{\frac{t}{2^n}}$ and therefore **to know t**...

  $\longrightarrow$ If number of iterations chosen too large, the success probability $\textbf{sin}\Big((2k+1)\theta\Big)^2$ 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 $\mathbf{x}$ $\left(\textit{i.e.}, f(\mathbf{x}) = 1\right)$ 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 $\mathbf{x}$ $\left(\textit{i.e.}, f(\mathbf{x}) = 1\right)$ 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 $\left(f(\mathbf{x}) = 1\right)$ 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 $|0^m\rangle$

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

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

Write:

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

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

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 **x** with probability p

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

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

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

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

$R_{|\psi\rangle} \neq O(n)$ quantum gates + 2 calls to $U = U_f \left(H^n \otimes I_2\right)$ which was designed to build
$$\frac{1}{\sqrt{2^n}} \sum_x |x\rangle\, |f(x)\rangle \ldots$$

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

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

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

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

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

$$\text{After amplitude amplification: } \left|\psi'\right\rangle \approx |\psi_V\rangle$$

$$\Big(\text{even equal with exact grover when amplitude } \alpha \text{ is known}\Big)$$

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 $\Big($think of R as the seed of a pseudo-random generator$\Big)$:

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

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

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 $\Big($think of $\mathbf{R}$ as the seed of a pseudo-random generator$\Big)$:

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

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

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

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

measuring outputs a solution with probability p

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

$\Big($the quantum algorithm finds a solution in time $\frac{\mathrm{Cost}(\mathcal{A})}{\sqrt{p}}$ instead of $\frac{\mathrm{Cost}(\mathcal{A})}{p}$ classically$\Big)$

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 $\mathbb{U} = \{z \in \mathbb{C} : |z| = 1\}$

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

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

Examples:

- $G = \mathbb{F}_2^n = \underbrace{\mathbb{F}_2 \times \cdots \times \mathbb{F}_2}_{n \text{ times}}$ with $\mathbb{F}_2$ binary field $\{0, 1\}$ embedded with $\oplus$ $\big($addition modulo 2$\big)$

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

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

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

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:

  $\left(\frac{\chi_x}{\sqrt{\sharp G}}\right)_{x\in G}$ is an 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

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

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

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

$$\begin{aligned} f_g : \widehat{G} &\longrightarrow \mathbb{C} \\ \chi &\longmapsto \chi(g) \end{aligned}$$

   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

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

$\longrightarrow$ Important concept in Simon's algorithm and Shor's algorithm! $\Big($see Lecture 4&6$\Big)$

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

Fourier transform:

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

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

Notice that:

$\widehat{f}(x) = \left\langle f, \frac{\chi_x}{\sqrt{\sharp G}} \right\rangle$ where $\langle \cdot, \cdot \rangle$ is the standard scalar product over functions, $\langle f, g \rangle \stackrel{\text{def}}{=} \sum_{x \in G} f(x)\overline{g}(x)$

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

Exercise:

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

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

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

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

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

Exercise:

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

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

Given $\mathbf{x}$, what is the cost for $\left(\text{classically}\right)$ computing $\widehat{f}(x)$?

COST FOR COMPUTING THE CLASSICAL FOURIER TRANSFORM

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

$\longrightarrow$ It costs $\sharp G$ $\Big($it is needed $\sharp G$ additions$\Big)$ ... Be careful: in practice $\sharp G = 2^n$

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

COST FOR COMPUTING THE CLASSICAL FOURIER TRANSFORM

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

$\longrightarrow$ It costs $\sharp G$ $\Big($it is needed $\sharp G$ additions$\Big)$ ... Be careful: in practice $\sharp G = 2^n$

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

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

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

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

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 $T(2^n) = O(2^n \underbrace{\log(2^n)}_{\text{rec. calls}}) = O\left(n2^n\right)$

FAST QUANTUM FOURIER TRANSFORM

Computing the quantum Fourier transform:

- QFT_G can be implemented in the quantum gate model in time $O\left(\log^3 \sharp G\right)$ for any finite Abelian group G
- $\mathrm{QFT}_{\mathbb{Z}/N\mathbb{Z}}$ can be implemented in time $O\left(\log^3 N\right)$ in the quantum gate model
- $\mathrm{QFT}_{\mathbb{Z}/2^n\mathbb{Z}}$ can be implemented in time $O\left(n^2\right)$ in the quantum gate model $\Big($here $n = \log 2^n = \log \sharp(\mathbb{Z}/2^n\mathbb{Z})\Big)$
- $\mathrm{QFT}_{\mathbb{Z}/2^n\mathbb{Z}}$ can be implemented up to some accuracy [a] in time $O\left(n \log n\right)$ in the quantum gate model
- $\mathrm{QFT}_{\mathbb{F}_2^n}$ can be implemented in time only $O(n)$ in the quantum gate model

[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 $\mathbb{Z}/2^n\mathbb{Z}$

A PARTICULAR CASE: HADAMARD TRANSFORM

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

$\longrightarrow$ 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 $\mathbb{F}_2^n$ $\left(\mathrm{QFT}_{\mathbb{F}_2^n}\right)$:

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

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

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

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

Gates that we will use:

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

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

Notation:

For any integer $j \in [\![0, 2^n - 1]\!]$, binary decomposition $j = j_1 \dots j_n$ where j_1 is the most significant bit

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

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

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

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

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

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

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

Idea: write the binary decomposition of $k\cdot 2^{-\ell}$

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

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

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

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

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

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

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

THE CASE 2^3

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

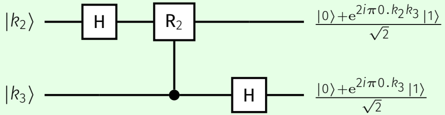
$$\left(\frac{|0\rangle + e^{2i\pi 0.k_3}|1\rangle}{\sqrt{2}}\right) \bigotimes \left(\frac{|0\rangle + e^{2i\pi 0.k_2k_3}|1\rangle}{\sqrt{2}}\right) \bigotimes \left(\frac{|0\rangle + e^{2i\pi 0.k_1k_2k_3}|1\rangle}{\sqrt{2}}\right)$$

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

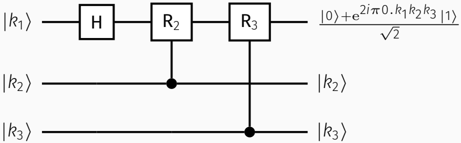
$$|k_3\rangle \xrightarrow{\mathrm{H}} \frac{|0\rangle + (-1)^{k_3}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + e^{2i\pi 0.k_3}|1\rangle}{\sqrt{2}} \qquad \left(0.k_3 = 0 \text{ if } k_3 = 0 \text{ or } \tfrac{1}{2} \text{ if } k_3 = 1\right)$$

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

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



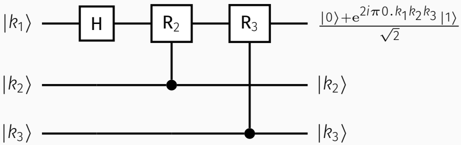
3. Sending $|k_3\rangle\,|k_2\rangle\,|k_1\rangle$ through the following circuit:



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

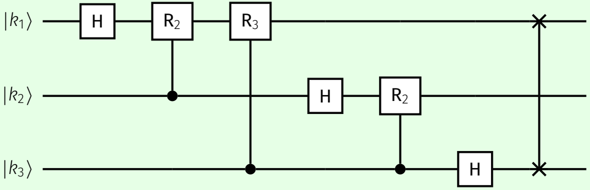
THE CASE 2^3

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



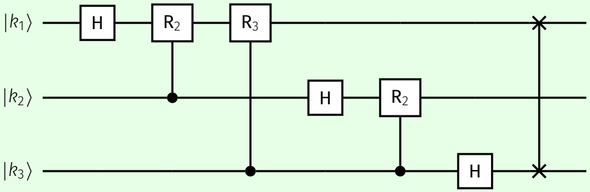
Combining this with the previous circuit gives almost the good state not in the good order: swap!

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: $O(n^2) + n \cdot \textbf{SWAP} = O(n^2) = O\Big(\left(\log 2^n\right)^2 \Big)$ gates

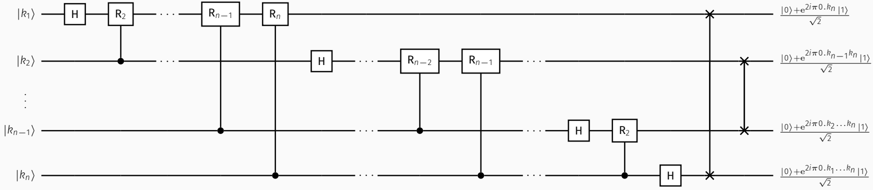
$\longrightarrow$ In particular gates $R_2, \ldots, R_n$ are used!

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 $s \geq C \log n$ $\Big($with C constant$\Big)$ will lead to the result with accuracy $\leq \frac{1}{n}$

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

GENERAL CASE: THE QUANTUM CIRCUIT



EXERCISE SESSION