

Day 2: Quantum Algorithms

Overview:

- I.) Deutsch-Jozsa algorithm: Oracles, DJ theory, implementation with Qiskit
- II.) Grover's algorithm: Grover theory, amplitude amplification, implementation with Qiskit

I. Deutsch-Jozsa algorithm

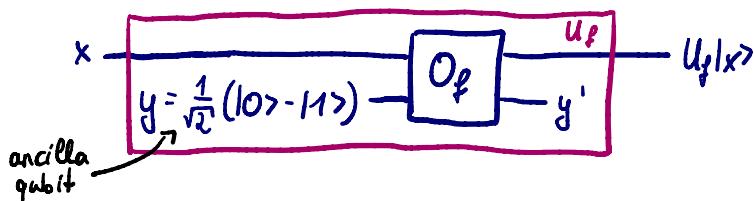
Oracles

- assume we have access to an oracle, e.g. a physical device that we cannot look inside, to which we can pass queries and which returns answers
 \Rightarrow goal: determine some property of the oracle using the minimal number of queries
- on a classical computer, such an oracle is given by a fct. $f: \underbrace{\{0,1\}^n}_{\text{input string}} \rightarrow \underbrace{\{0,1\}^m}_{\text{output string}}$
- on a quantum computer, the oracle must be reversible:

n qubits $\{x\}$ $\xrightarrow{O_f}$ x
 m qubits $\{y\}$ $\xrightarrow{O_f}$ $y \oplus f(x)$

O_f : bit oracle, can be seen as a unitary which performs the map $O_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle$

\rightarrow for $f: \{0,1\}^n \rightarrow \{0,1\}^m$, we can construct U_f :



$$O_f |x\rangle |y\rangle = \frac{1}{\sqrt{2}} (|x\rangle |0\rangle - |x\rangle |1\rangle) \xrightarrow{O_f} \frac{1}{\sqrt{2}} (|x\rangle |0\rangle \oplus f(x)) - |x\rangle |1\rangle \oplus f(x)) \xrightarrow{U_f} \begin{cases} \frac{1}{\sqrt{2}} |x\rangle (|0\rangle - |1\rangle) = |x\rangle |y\rangle, & \text{if } f(x)=0 \\ \frac{1}{\sqrt{2}} |x\rangle (|1\rangle - |0\rangle) = -|x\rangle |y\rangle, & \text{if } f(x)=1 \end{cases}$$

$$= (-1)^{f(x)} |x\rangle |y\rangle$$

\Rightarrow indep. of $|y\rangle \Rightarrow U_f$: phase oracle, which performs the map $U_f |x\rangle = (-1)^{f(x)} |x\rangle$

Hadamard on n qubits: recall that $H|0\rangle = |+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $H|1\rangle = |- \rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$

$$\Rightarrow \text{for } x \in \{0,1\}^n: |x\rangle \xrightarrow{\boxed{H}} |y\rangle = \frac{1}{\sqrt{2^n}}(|0\rangle + (-1)^{x_1}|1\rangle) = \frac{1}{\sqrt{2^n}}(-1)^{x_1}|0\rangle + (-1)^{x_1}|1\rangle = \frac{1}{\sqrt{2^n}} \sum_{k \in \{0,1\}^n} (-1)^{k \cdot x} |k\rangle$$

$$\Rightarrow \text{for } x \in \{0,1\}^n: |x\rangle \left(\begin{array}{c} |x_0\rangle \xrightarrow{\boxed{H}} |y_0\rangle \\ |x_1\rangle \xrightarrow{\boxed{H}} |y_1\rangle \\ \vdots \\ |x_{n-1}\rangle \xrightarrow{\boxed{H}} |y_{n-1}\rangle \end{array} \right) |y\rangle = H^{\otimes n}|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{k \in \{0,1\}^n} (-1)^{k \cdot x} |k\rangle$$

inner product
 \downarrow
 $k \cdot x$

↳ every $|y_i\rangle$ is either $|+\rangle$ or $|-\rangle$
 $\Rightarrow |y\rangle$ must be a superposition of all possible 2^n bit strings

e.g. $|x\rangle = |01\rangle$:

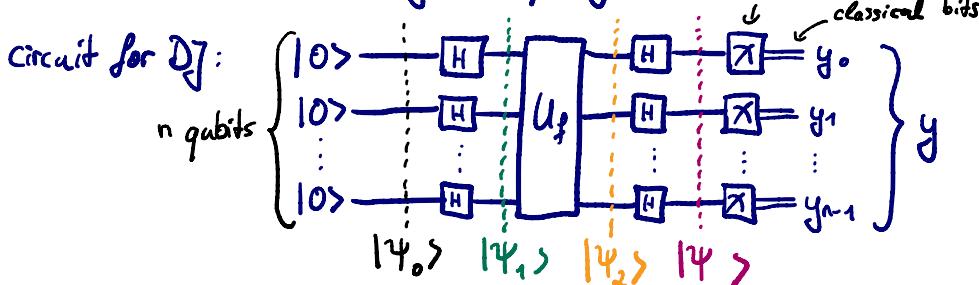
$$\left. \begin{array}{c} |0\rangle \xrightarrow{\boxed{H}} |+\rangle \\ |1\rangle \xrightarrow{\boxed{H}} |-\rangle \end{array} \right\} |y\rangle = |+\rangle \otimes |-\rangle = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)$$

Deutsch-Jozsa algorithm

- We are given a function $f: \{0,1\}^n \rightarrow \{0,1\}$, realized by an oracle, of which we know that it is either constant (\Rightarrow all inputs map to the same output) or balanced ($\#$ inputs that map to '0' and '1' is equal)
- Goal: Determine whether f is constant or balanced
- classical solution: we need to ask the oracle at least twice, but if we get twice the same output, we need to ask again, ...
 \rightarrow at most $\frac{N}{2} + 1 = 2^{n-1} + 1$ queries, $n: \#$ input bits, $N = 2^n: \#$ realizable bit strings

demonstrative example: 2^n different ways to throw a coin \rightarrow is the coin fair?

- quantum solution: needs only one query!



Claim: If the outcome y equals the bitstring $00\ldots 0$, then f is constant, otherwise it is balanced

Proof: Let us check the state after every step:

$$\begin{aligned}
 & \cdot |\Psi_0\rangle = |00\dots 0\rangle = |0\rangle^{\otimes n} \\
 & \cdot |\Psi_1\rangle = H^{\otimes n}|\Psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} \underbrace{(-1)^{\underset{x \cdot \Psi_0}{\text{}}}_{=+1}}_{\text{linearity}} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle \\
 & \cdot |\Psi_2\rangle = U_f |\Psi_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} U_f |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} |x\rangle \\
 & \cdot |\Psi_3\rangle = H^{\otimes n} \cdot |\Psi_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} \cdot H^{\otimes n} |x\rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x)} \cdot \sum_{k \in \{0,1\}^n} (-1)^{k \cdot x} |k\rangle \\
 & = \sum_{k \in \{0,1\}^n} \left[\frac{1}{2^n} \sum_{x \in \{0,1\}^n} (-1)^{f(x) + k \cdot x} \right] |k\rangle = : c_k |k\rangle
 \end{aligned}$$

\Rightarrow probability to measure the zero-string $|00\dots 0\rangle$:

$$\begin{aligned}
 P[y=00\dots 0] & \stackrel{\text{Born rule}}{=} |\langle 00\dots 0 | \Psi_3 \rangle|^2 = \left| \sum_{k \in \{0,1\}^n} c_k \cdot \underbrace{\langle 00\dots 0 | k \rangle}_{\substack{=1, \text{ if } k=00\dots 0 \\ =0, \text{ else (orthogonal)}}} \right|^2 = |c_{00\dots 0}|^2 \\
 & = \left| \frac{1}{2^n} \cdot \underbrace{\sum_{x \in \{0,1\}^n} (-1)^{f(x)}}_{\substack{\text{if } f \text{ const.} \\ \text{if } f \text{ balanced}}} \right|^2 = \begin{cases} 1, & \text{if } f \text{ const.} \\ 0, & \text{if } f \text{ balanced} \end{cases} \\
 & = \begin{cases} +2^n, & \text{if } f(x) \equiv 0 \\ -2^n, & \text{if } f(x) \equiv 1 \\ 0, & \text{if } f \text{ balanced} \end{cases}
 \end{aligned}$$

□

II. Grover's algorithm

- algorithm "searching an unsorted database" with $N=2^n$ elements in $\Theta(\sqrt{N})$ time rather: find x s.t. $f(x)=1$

→ classical alg. needs on average $\frac{N}{2} = \Theta(N)$ time

- goal: find ω , given an oracle U_f with $f: \{0,1\}^n \rightarrow \{0,1\}$, $f(x) = \begin{cases} 1, & \text{if } x=\omega \\ 0, & \text{else} \end{cases}$, $f_0(x) = \begin{cases} 0, & \text{if } x=0..0 \\ 1, & \text{else} \end{cases}$

$$\text{phase oracle: } U_f(x) = (-1)^{f(x)} = |x\rangle \Rightarrow U_f: |\omega\rangle \rightarrow -|\omega\rangle$$

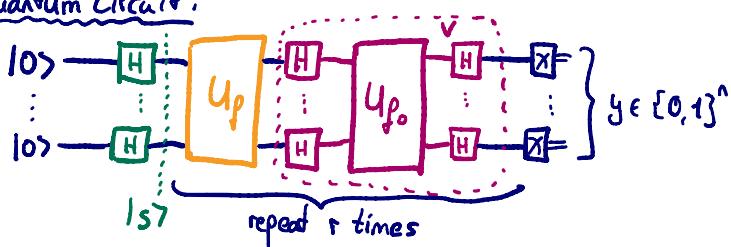
$$|x\rangle \rightarrow |x\rangle$$

$$Vx + \omega$$

$$U_f: |0\rangle^{\otimes n} \rightarrow |0\rangle^{\otimes n}$$

$$|x\rangle \rightarrow -|x\rangle \quad \forall x \neq \omega..0$$

• quantum circuit:



$$\hookrightarrow U_f = 11 \cdot 2| \omega \rangle \langle \omega |$$

$$\hookrightarrow U_{f_0} = 2|0\rangle \langle 0|^{\otimes n} - 11$$

Claim: $y = \omega$ (with high prob.)

Proof: Let us define the uniform superposition state $|s\rangle := H^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle$

$$\text{and } V := H^{\otimes n} \cdot U_{f_0} \cdot H^{\otimes n} = H^{\otimes n} \cdot 2|0\rangle \langle 0|^{\otimes n} \cdot H^{\otimes n} - H^{\otimes n} \cdot H^{\otimes n} = 2|s\rangle \langle s| - 11$$

⇒ Grover's algorithm carries out the operation $(V \cdot U_f)^r$ on the state $|s\rangle$.

Let Σ be the plane spanned by $|s\rangle$ and $|\omega\rangle$ and let $|\omega^\perp\rangle$ be the state

$$\text{orthogonal to } |\omega\rangle \text{ in } \Sigma: \quad |\omega^\perp\rangle := \frac{1}{\sqrt{2^n-1}} \sum_{x \neq \omega} |x\rangle$$

$$\Rightarrow |s\rangle = \sqrt{\frac{2^n-1}{2^n}} |\omega^\perp\rangle + \frac{1}{\sqrt{2^n}} |\omega\rangle = \cos \frac{\theta}{2} |\omega^\perp\rangle + \sin \frac{\theta}{2} |\omega\rangle$$

$$\begin{aligned} \text{define } \theta \text{ s.t. } \sin \frac{\theta}{2} &= \frac{1}{\sqrt{2^n}} \\ \Rightarrow \theta &= 2 \cdot \arcsin \frac{1}{\sqrt{2^n}} \end{aligned}$$

protocol:

1.) Prepare $|s\rangle$

2.) Apply $U_f = 11 \cdot 2| \omega \rangle \langle \omega |$ → reflection at $|\omega^\perp\rangle$

3.) Apply $V = 2|s\rangle \langle s| - 11$ → reflection at $|s\rangle$

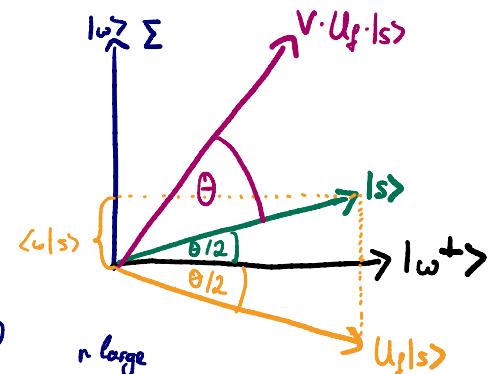
⇒ $V \cdot U_f$ corresponds to a rotation by an angle θ

⇒ after r applications of 2) & 3), the state is rotated by $r \cdot \theta$

$$\hookrightarrow \text{choose } r, \text{ s.t. } r \cdot \theta + \frac{\theta}{2} \approx \frac{\pi}{2} \Rightarrow r = \frac{\pi}{2\theta} - \frac{1}{2} = \frac{\pi}{4 \cdot \arcsin \frac{1}{\sqrt{2^n}}} - \frac{1}{2} \approx \frac{\pi}{4} \sqrt{2^n} = O(\sqrt{N})$$

⇒ after r calls to the oracle, the final meas. will result in state $|\omega\rangle$ with min. probability

$$p(\omega) \geq 1 - \sin^2 \frac{\theta}{2} = 1 - \frac{1}{2^n} \quad (\text{if } \xrightarrow{\text{Up}} |\omega^\perp\rangle)$$

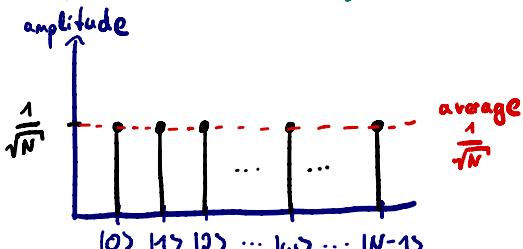


□

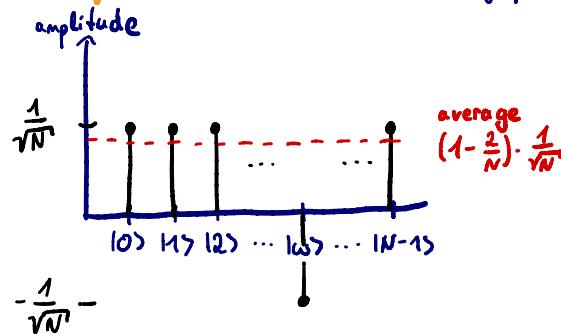
Amplitude amplification

The general idea behind Grover's algorithm is amplitude amplification. Let us have a look at the amplitudes at each step in Grover's algorithm:

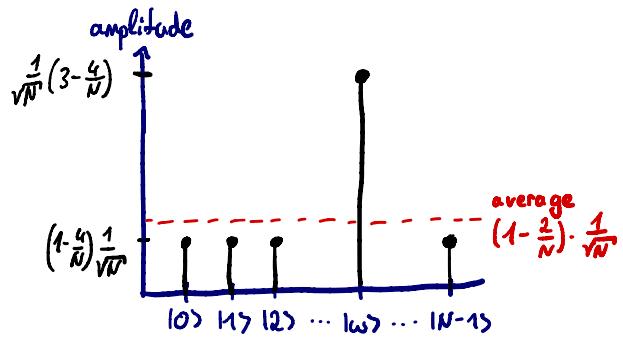
$$1.) |s\rangle := H^{\otimes n} |0\rangle^{\otimes n}$$



$$2.) U_f |s\rangle = (1 - 2|w\rangle\langle w|) |s\rangle \rightarrow \text{flip amplitude of } |w\rangle$$



$$3.) V \cdot U_f \cdot |s\rangle = (2|s\rangle\langle s| - 1) \cdot U_f \cdot |s\rangle \rightarrow \text{reflect amplitude about the average amplitude}$$



As for $|q\rangle := \sum_i \alpha_i |i\rangle \quad V|q\rangle$ gets:

$$(2|s\rangle\langle s| - 1)|q\rangle = 2 \cdot \frac{1}{N} \cdot \sum_j |j\rangle \cdot \sum_k \langle k| \cdot \sum_i \alpha_i |i\rangle - \sum_i \alpha_i |i\rangle$$

$$= 2 \cdot \frac{\sum_k \alpha_k}{N} \cdot \sum_j |j\rangle - \sum_j \alpha_j |j\rangle$$

$$= \sum_j (2 \cdot \langle \alpha \rangle - \alpha_j) |j\rangle$$

Reflection of α_j about average $\langle \alpha \rangle$ (\rightarrow if $\alpha_j = \langle \alpha \rangle + \Delta$ then $\alpha_j' = 2\langle \alpha \rangle - \alpha_j - \langle \alpha \rangle - \Delta$)

\Rightarrow by repeating step 2) & 3), the amplitude of $|w\rangle$ will increase further \Rightarrow amplitude amplification!

Multiple marked elements

When we have M marked elements w_i , we define the winning state as

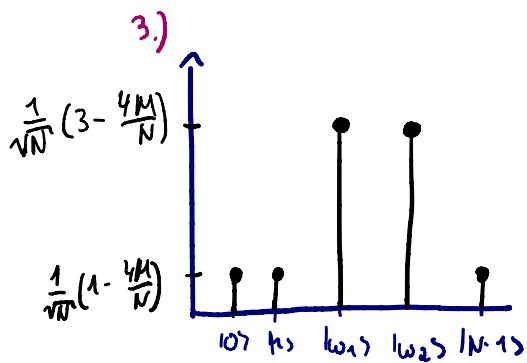
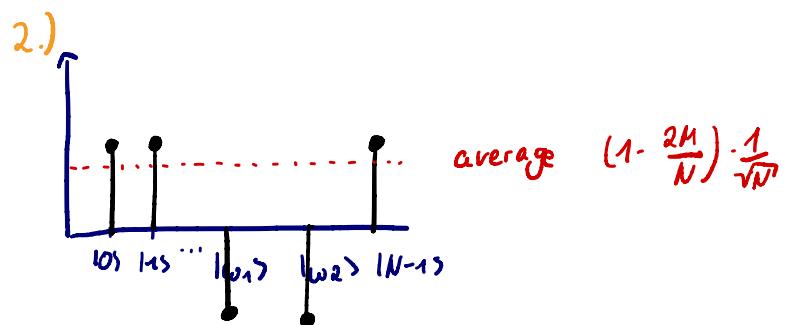
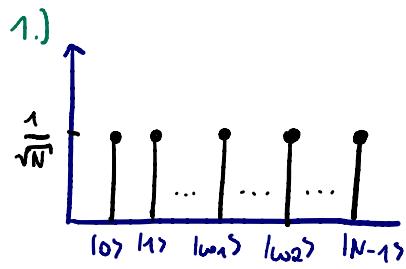
$$|w\rangle := \frac{1}{\sqrt{M}} \sum_{i=1}^M |w_i\rangle \quad \rightarrow \quad |w^\perp\rangle = \frac{1}{\sqrt{N-M}} \cdot \sum_{x \notin \{w_1, \dots, w_M\}} |x\rangle$$

$$\Rightarrow |s\rangle = \frac{\sqrt{N-M}}{\sqrt{N}} \cdot |w^\perp\rangle + \sqrt{\frac{M}{N}} \cdot |w\rangle =: \cos \frac{\theta}{2} |w^\perp\rangle + \sin \frac{\theta}{2} |w\rangle$$

$$\hookrightarrow \sin \frac{\theta}{2} = \sqrt{\frac{M}{N}} \quad \rightarrow \text{angle becomes larger!}$$

$$\Rightarrow r = \frac{\pi}{4 \cdot \arcsin(\sqrt{\frac{M}{N}})} - \frac{1}{2} = O(\sqrt{\frac{N}{M}})$$

→ we can see this speedup also when looking at the amplitudes:



⇒ goes faster