Unidade: Journeys into Theoretical physics 2024
Disciplina: 4 lectures on quantum computing: Lecture 1 part 1
Docente:
Aluno:
RA: Período: 22 - 26/07/2024 Data: / /

Lecture 1: The framework of quantum computing

- the main idea: "can we use quantum mechanics to solve computational problems?"

- Why?: "Quantum advantage" → $\boxed{\text{making formidable Tasks more manageable}}$ ← this is the key point for the entire business.

- HOW?:

classical computing → classical BIT $\{0,1\}$ → $\boxed{\text{classical physics}}$

$\Rightarrow$

Quantum computing → Qubit $\{|0\rangle, |1\rangle\}$ → $\boxed{\text{Quantum Physics}}$

Plan of Today:

- Refresh of Q.M.: bits vs qubits → states, Measurement, Evolution

- The circuit model: gates

- Entanglement & some applications: → Teleportation, (Quantum game), (superdense coding)

- Quantum ~~computing~~ Mechanics refreshments

number of bits

- classical information : it is based on bits $\{0,1\}^N$
  - → 010, 011, 001 → 8 possible states

- Quantum information:
  - → $\cancel{\{0,1\}}$ ⟹ "qubit": a 2 level system ⟹ $\boxed{\{|0\rangle, |1\rangle\}}$
  - → The main quantum feature: **SUPERPOSITION PRINCIPLE**

"classical states"

$$\boxed{|\psi\rangle = \alpha|0\rangle + \beta|1\rangle}$$

"Amplitudes" → $\alpha, \beta \in \mathbb{C}$ s.t. $|\alpha|^2 + |\beta|^2 = 1$

⟹ BLOCH SPHERE

$$\boxed{|\psi\rangle = \begin{pmatrix}\alpha \\ \beta\end{pmatrix}}$$

⟹ a much richer setup ! ⟹ 3 real numbers

How to deal with multiple qubits ??

→ Tensor product ⟹ $\mathcal{H}_2 = \mathcal{H}_1 \otimes \mathcal{H}_1 = \mathcal{H}^{\otimes 2}$

⟹ possible "classical states":

$|ij\rangle$ are orthonormal ; $\langle ij|kl\rangle = \delta_{ik}\delta_{jl}$

$$\left.\begin{matrix}|00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle\end{matrix}\right\} \Rightarrow |\psi\rangle = \alpha_0|00\rangle + \alpha_1|01\rangle + \alpha_2|10\rangle + \alpha_3|11\rangle$$

$$|\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2 + |\alpha_3|^2 = 1$$

in general: n qubit ⟹ $N = 2^n$ "classical states" $|\psi\rangle = \begin{pmatrix}\alpha_0 \\ | \\ \alpha_{N-1}\end{pmatrix}$ ← normali

Now we have a state, $\boxed{\text{What do we do with it?}}$ — Measure it!
— Evolve it!

⇒ $\boxed{\text{Measurements}}$ (there is much more to say about it, here we just consider the simplest case).

→ $\boxed{\text{Measurements in the computational basis}}$

suppose we have: $\boxed{\underbrace{|\psi\rangle = a_0|00\rangle + a_1|01\rangle + a_2|10\rangle + a_3|11\rangle}_{\text{superposition}}}$

⇒ If we measure the state → we will see a classical OUTCOME

⇒ we will see one among $\left\{\begin{matrix}|00\rangle \\ |01\rangle \\ |10\rangle \\ |11\rangle\end{matrix}\right\}$ which one?
↳ Probabilistic

$P(|00\rangle) = |a_0|^2$ and so on ⇒ we see why $\boxed{\sum_i |a_i|^2 = 1}$

we have to observe something

notice: a_i are NOT probabilities
↳ $|a_i|^2$ are probabilities

less information (you miss the phases)

⇒ In a sense: All the Magic of QC is on this simple fact: Q.C. is more than Probabilistic computing exactly for the Presence of these phases.

Evolution of states

operation (linear, given the axioms of quantum mechanics)

$$|\psi\rangle = \alpha_0|0\rangle + \_\_ + \alpha_{N-1}|N-1\rangle \xRightarrow{\hat{U}} |\psi'\rangle = \beta_0|0\rangle + \_\_ + \beta_{N-1}|N-1\rangle$$

which restriction on $\hat{U}$?

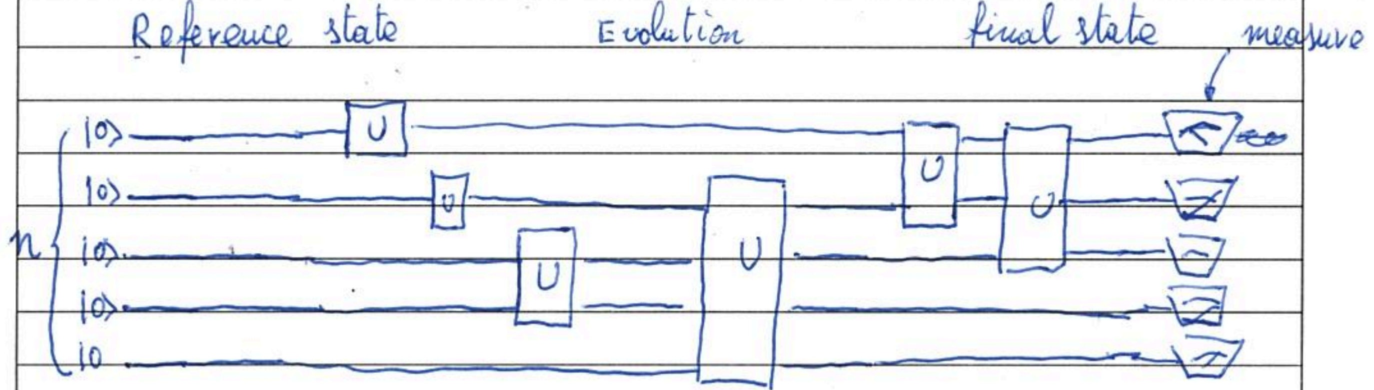
→ must preserve probability : $\sum_i |\beta_i|^2 = 1$

$$\Rightarrow \boxed{\hat{U} \text{ must be unitary}} \ \boxed{UU^\dagger = \mathbb{1}}$$

CIRCUITS & GATES

to represent basic unitaries (

→ time evolution



$\boxed{U}$: are unitary operators, called "GATES"

- typically 1-qubit (X, Y, Z, HADAMARD)
  2-qubits (CNOT, CZ)
  3-qubits (Toffoli)

Unidade: Journeys into theoretical physics 2024
Disciplina: 4 lectures on quantum computing: Lecture 1 part 2
Docente:
Aluno:
RA: Período: 22÷26/07/2024 Data: / /

- some gates

X-Gate: is the Pauli σ^x: Flips the qubit state

$$X|a\rangle = |1-a\rangle \qquad -\boxed{X}- \qquad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$
$a = 0, 1$

more general

Z-gate: is $Z|a\rangle = (-1)^a |a\rangle$ $\quad -\boxed{Z}- \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \rightarrow \boxed{R_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{-i\phi} \end{pmatrix}}$

Hadamard

H-Gate: it acts like $H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \equiv |+\rangle$

$-\boxed{H}-$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \equiv |-\rangle$$

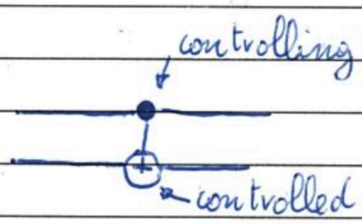
$$H \equiv \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Notice:
$X|\pm\rangle = \pm|\pm\rangle$

→ this is probably the most important 1-Qubit gate:

→ It creates superposition $\quad$ CLASSICAL $\xrightarrow{H}$ QUANTUM

- 2-Qubit: CNOT



→ $CNOT\,|0\rangle|a\rangle = |0\rangle|a\rangle$
$CNOT\,|1\rangle|a\rangle = |1\rangle|1-a\rangle$

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

This gate creates entanglement

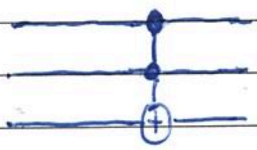
similarly: CZ gate $\quad CZ = \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & -1 \end{pmatrix}$

$$CZ\,|0\rangle|a\rangle = |0\rangle|a\rangle$$
$$CZ\,|1\rangle|a\rangle = |1\rangle|1-a\rangle$$

- 3-qubits gate : the Toffoli gate (CCNOT)

↳ it's like the CNOT but with (2) control gates

it flips the third qubit if both the controling qubits are $|1\rangle$



→ this gate is essential to simulate classical functions:

$$\boxed{f(x) : \{0,1\}^n \longrightarrow \{0,1\}}$$ (More next lecture)

■ Entanglement : (correlations between different qubits)

can be thought as: SUPERPOSITION ⊕ Multiple qubits

↳ suppose the system is made by 2 parts A and B ⇒ $|\psi\rangle$ is entangled if I cannot write it as $\boxed{|\psi\rangle_A \otimes |\psi_B\rangle}$

Example :

• $|\psi\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle + |00\rangle\right) = |0\rangle \otimes \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) = |0+\rangle$

↳ Not Entangled

• $|\psi\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) \neq |\psi_A\rangle \otimes |\psi_B\rangle$

$|||$

$|EPR\rangle$ (correlation) $\quad \Rightarrow \boxed{\text{Entangled}}$

Entanglement is at the core of Q.C. : APPLICATIONS

■ Quantum teleportation :

suppose we have 2 persons: Alice
Bob

|classical chanel| → they can send each other classical bits here

$A \longleftrightarrow B$

A has a qubit : $\boxed{|\psi\rangle = \alpha|0\rangle + \beta|1\rangle}$

→ she wants to send it to B, how to do that?

- Classically: IMPOSSIBLE : α, β may require infinite Bits

Ex: $\alpha = \beta = \frac{1}{\sqrt{2}}$

- Quantum: suppose they share $\boxed{|EPR\rangle = \frac{1}{\sqrt{2}}\left(|\underset{A}{0}\underset{B}{0}\rangle + |\underset{A}{1}\underset{B}{1}\rangle\right)}$

→ Now they can ⁈

Let'see How:

- total state $|\psi_T\rangle = |\psi\rangle \otimes |EPR\rangle = (\alpha|0\rangle + \beta|1\rangle) \otimes \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

- Alice does the Following:

1) CNOT on her 2 qubits 2) H on the first

⇒ see what happens:

$$|\psi_T\rangle = \frac{\alpha}{\sqrt{2}}|000\rangle + \frac{\alpha}{\sqrt{2}}|011\rangle + \frac{\beta}{\sqrt{2}}|100\rangle + \frac{\beta}{\sqrt{2}}|111\rangle$$

$$CNOT|\psi_T\rangle = \frac{\alpha}{\sqrt{2}}|000\rangle + \frac{\alpha}{\sqrt{2}}|011\rangle + \frac{\beta}{\sqrt{2}}|110\rangle + \frac{\beta}{\sqrt{2}}|101\rangle$$

- $H_1 CNOT|\psi_T\rangle = \frac{\alpha}{2}\big(\underset{x}{|0\rangle}+|1\rangle\big)|00\rangle + \frac{\alpha}{2}\big(\underset{\sim}{|0\rangle}+|1\rangle\big)|11\rangle + \frac{\beta}{2}\big(\underset{\sim}{|0\rangle}-|1\rangle\big)|10\rangle +$

$+\frac{\beta}{2}\big(\underset{x}{|0\rangle}-|1\rangle\big)|01\rangle$

$= \frac{1}{2}|00\rangle\big(\alpha|0\rangle+\beta|1\rangle\big) + \frac{1}{2}|01\rangle\big(\alpha|1\rangle+\beta|0\rangle\big) + \frac{1}{2}|10\rangle\big(\alpha|0\rangle-\beta|1\rangle\big) +$

$+\frac{1}{2}|11\rangle\big(\alpha|0\rangle-\beta|1\rangle\big)$

- we start to see an interesting pattern: the superposition is now "in Bob's hands"

■ A measures her qubits $\Rightarrow$ she gets a "classical state" $|a\, b\rangle$

■ A communicates (a, b) to B:

B: if $b = 1$: B applies X-gate
↓
if $a = 1$: B " Z-gate.

$\Rightarrow$ $\boxed{\text{Now B has } |\psi\rangle = \alpha|0\rangle+\beta|1\rangle}$

Teleportation

unesp UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"

Unidade Journeys into Theoretical physics 2024
Disciplina 4 lectures on Quantum computing: Lecture 2, part 1
Docente
Aluno
RA Período 22÷26/07/2024 Data / /

Lecture 2: more about circuits, The query model, some algorithms

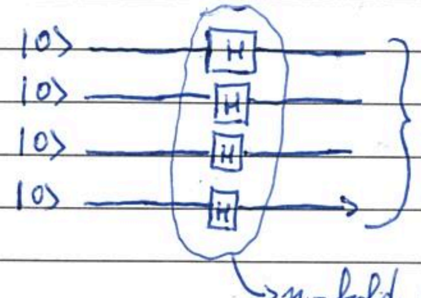
more about circuits	The query model	some algorithms
1) the n-fold Hadamard 2) classical functions with toffoli. 3) 1)+2) → Quantum parallelism	1) a high-level view on circuits 2) the standard query 3) the phase query	1) Deutsch-Josza 2) Bernstein-Vazirani ~~3) Quantum Fourier Transform / phase estimation~~

The n-fold Hadamard

→ this is a very used step in Many algorithms → often in inizialization

let us consider the following n-qubit circuit

$$H^{\otimes n}|0^n\rangle = |\psi\rangle$$



$|\psi\rangle$ what is this state?

→ n-fold Hadamard

$|\psi\rangle = |+\rangle \otimes |+\rangle \otimes |+\rangle \otimes |+\rangle \ldots$

it's a product state (No entanglement) but

$$\boxed{|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{j\{0,1\}^n} |j\rangle = \frac{|0000\rangle + |0001\rangle + \ldots + |1111\rangle}{\sqrt{2^n}}}$$

→ it creates a uniform superposition of all n-bit strings.

what happens if $|0^n\rangle \longrightarrow |i\rangle$

$\quad\quad$ generic bit-string $= i \in \{0,1\}^n$

$$\Rightarrow \boxed{H^{\otimes n}|i\rangle = \frac{1}{\sqrt{2^n}} \sum_{j\in\{0,1\}^n} (-1)^{i\cdot j} |j\rangle \qquad i\cdot j = \sum_k i_k j_k}$$

Classical Boolean functions with Toffoli

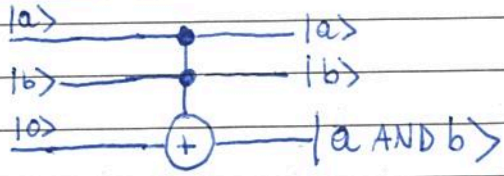
→ Boolean functions:

$$\boxed{f(x): \{0,1\}^n \longrightarrow \{0,1\}}$$

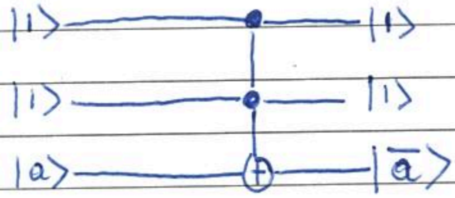
> **Theorem**: any classical ~~for~~ boolean function can be computed with ~~two~~ AND and NOT gates only
>
> we are not going to prove it!

Toffoli gate:

→ AND gate →



→ NOT gate →



$\Rightarrow$ We see that classical boolean functions can be computed by means of Toffoli circuits only

Quantum parallelism

- let's consider the circuit $|i \in \{0,1\}^n\rangle|0\rangle \xrightarrow{U} |i\rangle|f(i)\rangle$

$\Rightarrow$ by combining $H^{\otimes n}$ with U we get

$$\Rightarrow UH^{\otimes n}\left(|0^n\rangle|0\rangle\right) = U\left(\frac{1}{\sqrt{2^n}}\sum_{i\in\{0,1\}^n} |i\rangle|0\rangle\right) = \frac{1}{\sqrt{2^n}}\sum_{i\in\{0,1\}^n} |i\rangle|f(i)\rangle$$

↳ Quantum parallelism: we just applied U once and we got all possible outcomes!

↳ ~~Is this useful~~ Is this useful?: as it is, NOT MUCH: if we measure the state we just get one randomized instance of $f(i)$

$\Rightarrow$ We will need more (namely ~~sup~~ interference & entanglement)

The query model:

↳ this is a model which allows a High-level description of quantum algorithms.

Setup: 1) an $N \equiv 2^n$-bit input ("oracle", "memory")

$$x = (\underbrace{x_0, x_1, \ldots, x_{N-1}}) \in \{0,1\}^N$$

↳ they can be labelled with n-bit strings

↳ they can be realized as $\{f(0), f(1) \ldots f(N-1)\}$ $\parallel$ via Toffoli circuits

2) a memory access:

↳ this is done via a unitary "black-Box"

$$O_x : |\overbrace{i, 0}^{n+1 \text{ qubits}}\rangle \longrightarrow |i, x_i\rangle$$

address, target

3) Making the memory access unitary

$$\boxed{O_x : |i, b\rangle \longmapsto |i, b \oplus x_i\rangle}$$ "standard" query

1 application of O_x is a "<u>QUERY</u>"

↳ often, O_x is the heaviest part to compute

$\Rightarrow$ **Query complexity**: How many times do we need to call O_x?

Another possibility: "<u>phase Query</u>"

let's apply $O_x : |i, -\rangle = O_x\left(|i\rangle \otimes \left(\frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right)\right)\right) =$

$$= \frac{1}{\sqrt{2}}\left(|i\rangle|x_i\rangle - |i\rangle|1 \oplus x_i\rangle\right) = (-1)^{x_i}|i\rangle|-\rangle$$

$\Rightarrow$ **Phase query**: $O_{x\pm}|i\rangle = (-1)^{x_i}|i\rangle$

Notice: $O_x|i, +\rangle = |i, +\rangle \Rightarrow O_x$ does nothing ~~de~~ this case

III Some Algorithms

we now have finished ~~the~~ with the "technology"

→ From now on, until the end of the course, we will discuss <u>Algorithms</u>.

<u>standard setup</u>:

Problem to solve → Classical solution (complexity) → Quantum Advantage

Problem to solve → Quantum solution (complexity) → Quantum Advantage

Unidade: Journeys into theoretical physics 2024
Disciplina: 4 lectures on Quantum computing: lecture 2, part 2
Docente:
Aluno:
RA: Período: 22÷26/07/2024 Data: / /

The Deutsch-Josza algorithm

- Problem: we have an N-bit string ($N=2^n$)

$$x = x_0 x_1 \dots x_{N-1}$$

and we know that 1 of these 2 options is satisfied:

(a) All x_i are EQUAL ("constant" case)

(b) $\frac{N}{2}$ x_is are $=0$
$\frac{N}{2}$ x_is " $=1$
("balanced" case)

→ We want to discriminate these 2 cases.

- Classical Deterministic: in the worst case scenario we need

$$O\left(\frac{N}{2}+1\right) \sim \boxed{O(2^n) \text{ queries}}$$

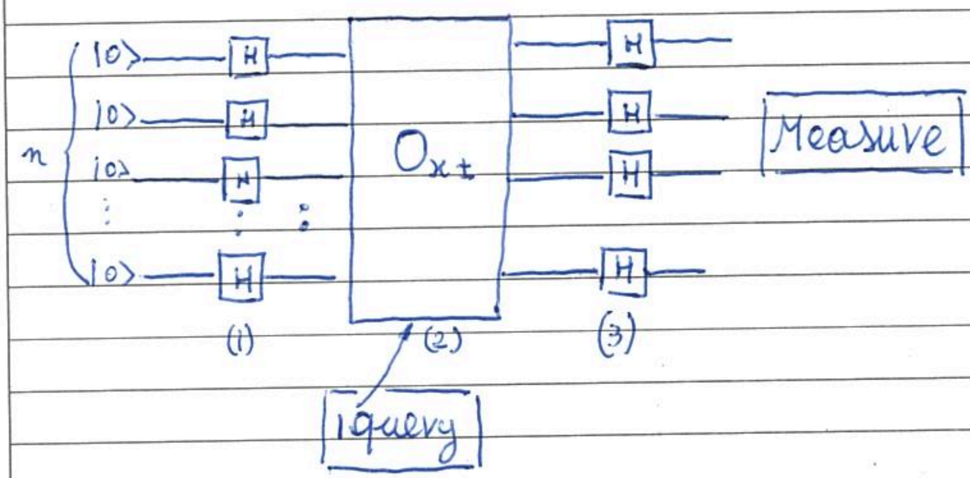
if we see $\frac{N}{2}$ consecutive 0s, still the output is undetermined

Probabilistic: $\boxed{O(1) \text{ queries are enough}}$

Extract randomly at few positions and check: if all Equals, probably "constant case"

⇒ Classical summary:
→ Deterministic inefficient
→ Probabilistic efficient

Quantum consider the following circuit



and lets follow what is going to happen

$$(1):\ |\psi_{(1)}\rangle = \frac{1}{\sqrt{2^n}} \sum_{i\in\{0,1\}^n} |i\rangle$$

$$(2):\ |\psi_{(2)}\rangle = \frac{1}{\sqrt{2^n}} \sum_{i\in\{0,1\}^n} (-1)^{x_i} |i\rangle$$

$$(3)\ |\psi_{(3)}\rangle = \frac{1}{2^n} \sum_{i\in(0,1\}^n} (-1)^{x_i} \sum_{j\in\{0,1\}^n} (-1)^{i\cdot j} |j\rangle$$

now $i\cdot 0 = 0 \ \forall i \in \{0,1\}^n$

$$\Rightarrow |\psi_{(3)}\rangle = \alpha_0 |0^n\rangle + \dots$$

$$\hookrightarrow \alpha_0 = \frac{1}{2^n} \sum_{i\in\{0,1\}^n} (-1)^{x_i} \begin{cases} +1 & \text{if } x_i = 0\ \forall i \\ -1 & \text{if } x_i = 1\ \forall i \\ 0 & \text{if } x \text{ is balanced} \end{cases}$$

$\Rightarrow$ after measurement
- $\to |0^n\rangle \Longrightarrow$ constant case
- any other state $\Rightarrow$ balanced case

$\Rightarrow$ Quantum summary: Deterministic with just 1 query

Bernstein Vazirani

Problem: $N = 2^n \quad x = x_0 x_1 \dots x_{N-1} \in \{0,1\}^N$

- x satisfies the property: ~~specifica~~

$$\boxed{\exists\, a \in \{0,1\}^n \text{ s.t. } x_i = (i \cdot a) \bmod 2 \ \forall i \in \{0,1\}^n}$$

- Task: find a

- Classical: no matter deterministic or randomized $\Rightarrow \boxed{O(n) \text{ queries}}$

Example: $\left.\begin{array}{l} x_{100\dots0} = a_0 \\ x_{01\dots0} = a_1 \\ \vdots \\ x_{00\dots1} = a_n \end{array}\right\} O(n)$ queries

- Quantum: We do Exactly the same as in JD algorithm

$x_i = (i \cdot a) \bmod \underbrace{2}_{\text{irrelevant}}$

$\Rightarrow$ ~~Final state~~ $|\Psi_{(2)}\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} (-1)^{x_i} |i\rangle = \frac{1}{\sqrt{2^n}} \sum_i (-1)^{i \cdot a} |i\rangle$

$\underset{\text{property of } H^{\otimes n}}{\Rightarrow}$ $\boxed{|\Psi_{(3)}\rangle = H^{\otimes n} |\Psi_{(2)}\rangle = |a\rangle}$

$\Rightarrow$ 1 Query + measurement and we get a

$\Rightarrow$ a $\boxed{\text{Polynomial advantage}}$

Unidade: Journeys into theoretical physics 2024
Disciplina: 4 lectures on quantum computing: Lecture 3, part 1
Docente
Aluno
RA Período: 22÷26/07/2024 Data / /

Lecture 3: 2 important algorithms: - Quantum Fourier transform
→ Phase estimation
- Grover search.

Quantum Fourier transform

- classical case first: the discrete Fourier transform (DFT)

- it's a $N \times N$ matrix F_N:

- F_N is unitary

- $|F_{N\,ab}|^2 = 1 \quad \forall a, b \in \{1, \dots, N\}$

⇒ Example: $\underbrace{F_2 = H}_{\text{Hadamard}} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$

for general $N \geq 2$: define $\omega_N = e^{2\pi i/N}$ (Nth root of unity)

$$\Rightarrow F_{N\,jk} = \frac{1}{\sqrt{N}}\,\omega_N^{jk} = \frac{1}{\sqrt{N}}\,e^{\frac{2\pi i}{N} j\cdot k} \rightarrow \boxed{F_N = \frac{1}{\sqrt{N}} \begin{array}{c|c} & k \\ \hline j & \begin{pmatrix} \cdots\cdots & \omega_N^{jk} \end{pmatrix} \end{array}}$$

F_N is unitary and symmetric $\Rightarrow F_N^{-1} = F_N^*$

- DFT is the map:

$$\text{DFT}: v_N \longrightarrow \hat{v}_N = F_N v_N \qquad (\hat{v}_N)_j = \frac{1}{\sqrt{N}} \sum_k \omega_N^{jk} (v_N)_k$$

- COMPLEXITY: $\boxed{O(N^2) \xrightarrow[\text{Fast Fourier transform}]{} O(N \log N)}$

The quantum case :

- F_N is a unitary matrix $\Rightarrow$ it defines a quantum operation

$$|\psi\rangle \longrightarrow \boxed{|\hat{\psi}\rangle = F_N |\psi\rangle}$$

N-qubit state

We look for an efficient implementation.

Notice : $F_N^{\text{classical}}$ and F_N^{quantum} are not the same ~~ve~~ operation

this acts with an actual vector, written on paper

this is a vector of amplitudes.

$\Rightarrow$ much like parallelism, we need to see what to do with it.

the Quantum circuit:

- Allowed gates : • $CR_s = \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^s} \end{pmatrix}}$ $\longrightarrow$ $R_1 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = Z$, $R_2 = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$

as usual controlled version

↳ this is just R_s

↳ for s large: $R_s \approx \mathbb{1}$

• Hadamard

- Linearity simplification :

↳ we just need to consider $F_N |k\rangle$

basis vector

$|k\rangle = |0\rangle, \_\_ |N-1\rangle$ with $N = 2^n$

$\Rightarrow |k\rangle$ is labelled by n-qubits

$|k\rangle = |k_1 \dots k_n\rangle$

most significant bit

now, notice the following property:

We have (base 2) $j = j_1 \ldots j_n$ $\Rightarrow$ $\boxed{0.j_1 \ldots j_n = \frac{j}{2^n} = \left(j_1 \cdot 2^{n-1} + j_2 \cdot 2^{n-2} + \ldots j_n 2^0\right) \cdot \frac{1}{2^n} = \sum_{l=1}^{n} j_l 2^{-l}}$

j_1: most significant

Example: $0.101 = \frac{5}{8} = 2^{-1} \cdot 1 + 2^{-2} \cdot 0 + 2^{-3} \cdot 1$

$\Rightarrow$ we can now compute $F_N |k\rangle$

$$F_N |k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \exp\left(2\pi i j k \cdot \frac{1}{2^n}\right) |j\rangle =$$

$$= \frac{1}{\sqrt{2^n}} \sum_{j \in \{0,1\}^n} \exp\left(2\pi i \sum_{l=1}^{n} j_l 2^{-l} k\right) |j_1 \ldots j_n\rangle =$$

$$= \frac{1}{\sqrt{2^n}} \sum_{j \in \{0,1\}^n} \prod_{l=1}^{n} e^{2\pi i j_l k / 2^l} |j_1 \ldots j_n\rangle =$$

$$\boxed{= \bigotimes_{l=1}^{n} \frac{1}{\sqrt{2}} \left(|0\rangle + e^{2\pi i k / 2^l} |1\rangle\right)}$$

important $\leftarrow$ no entanglement

now: $\exp\left(\frac{2\pi i k}{2^l}\right) = \exp\left(2\pi i \cdot (\underbrace{k_1 k_2 \ldots k_{n-l} . k_{n-l+1} \ldots k_n}_{\text{base 2}})\right)$

integer part $\rightarrow$ doesn't matter since $e^{2\pi i m} = 1$, m integer

$$= \exp\left(2\pi i \, 0.k_{n-l+1} \ldots k_n\right)$$

we are so ready for the circuit:

Example: $n=3 \rightarrow N=8$

$$F_8 |k_1 k_2 k_3\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\, 0.k_3}|1\rangle\right) \otimes \frac{1}{\sqrt{2}}\left(|0\rangle + e^{2\pi i\, 0.k_2k_3}|1\rangle\right) \otimes$$

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

now: $\cancel{\frac{1}{\sqrt{2}}}$ $|k_3\rangle$

$\rightarrow |0\rangle \longrightarrow \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$

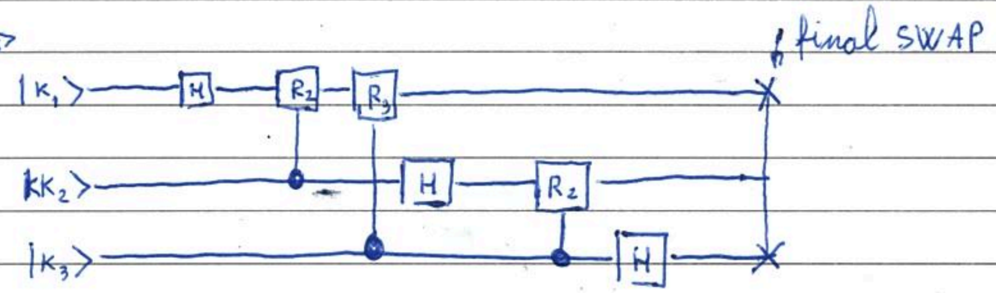
$\rightarrow |1\rangle \longrightarrow \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}} e^{2\pi i \overbrace{0.1}^{2^{-1}}}|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$

$\Rightarrow$ For $|k_3\rangle$ I simply need a Hadarmard

$\Rightarrow$ FOR $|k_2\rangle$ I need Hadarmard and then a $\overset{\text{with } k_3}{C}R_2$

$\Rightarrow$ For $|k_1\rangle$ " " " " a $\overset{\text{with } k_2}{C}R_2 + \overset{\text{with } k_3}{C}R_3$

$\Rightarrow$



Complexity $\longrightarrow$ at most n gates per qubit
$\longrightarrow$ n qubit
$\Longrightarrow O(n^2)$

$\longrightarrow$ "advantage" but remember that the two operat are not the same

$\longrightarrow$ Recall that classical $O(N \log N) = O(2^n \cdot n)$

Unidade: Journeys into theoretical physics 2024
Disciplina: 4 lectures on quantum computing: Lecture 3, part 2
Docente
Aluno
RA Período 22÷26/07/2024 Data / /

An application: Quantum phase estimation

Problem:
- We have a unitary U
- " " " eigenvector $|\psi\rangle$

$\Rightarrow U|\psi\rangle = \lambda |\psi\rangle$

$\hookrightarrow$ Unitary $\Rightarrow \lambda = e^{2\pi i \phi}$, $\phi \in [0,1)$

$\Rightarrow$ $\boxed{\text{Goal: estimate } \phi}$

Assumption: $\phi = 0.\underbrace{\phi_1 \phi_2 \ldots \phi_n}_{}$ n-bits precision (for simplicity)

- Algorithm:

1) start with $|0^n\rangle|\psi\rangle$

2) Apply $H^{\otimes n}$ on the first n-qubit $\Rightarrow |0^n\rangle|\psi\rangle \Rightarrow \frac{1}{\sqrt{2^n}} \sum_{j=0}^{N-1} |j\rangle|\psi\rangle$

3) Apply the map: $|j\rangle|\psi\rangle \to |j\rangle U^j|\psi\rangle = e^{2\pi i \phi j}|j\rangle|\psi\rangle$

$\Rightarrow$ the state is now $\left(\frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \exp(2\pi i \phi \cdot j)|j\rangle\right)|\psi\rangle = \left(F_N |2^n\phi\rangle\right)|\psi\rangle$

4) Apply F_N^{-1}: the state is $|\underbrace{2^n\phi}_{\phi_1 \ldots \phi_n}\rangle|\psi\rangle$

5) Measure the state $\Rightarrow \boxed{\phi_1 \ldots \phi_n}$

The Grover search algorithm (one of the most famous algorithms)

→ $N = 2^n$ we are given an $x = x_0 x_1 \ldots x_{N-1} \in \{0,1\}^N$

$\boxed{\text{Goal: Find } i \text{ s.t. } x_i = 1 \ (\text{if } \exists)}$

- First classical Complexity: $\Theta(N)$ (both $O(N)$ and $\Omega(N)$)

Now quantum:

def: t is the number of i s.t. $x_i = 1$ ⇒ Hamming Weight

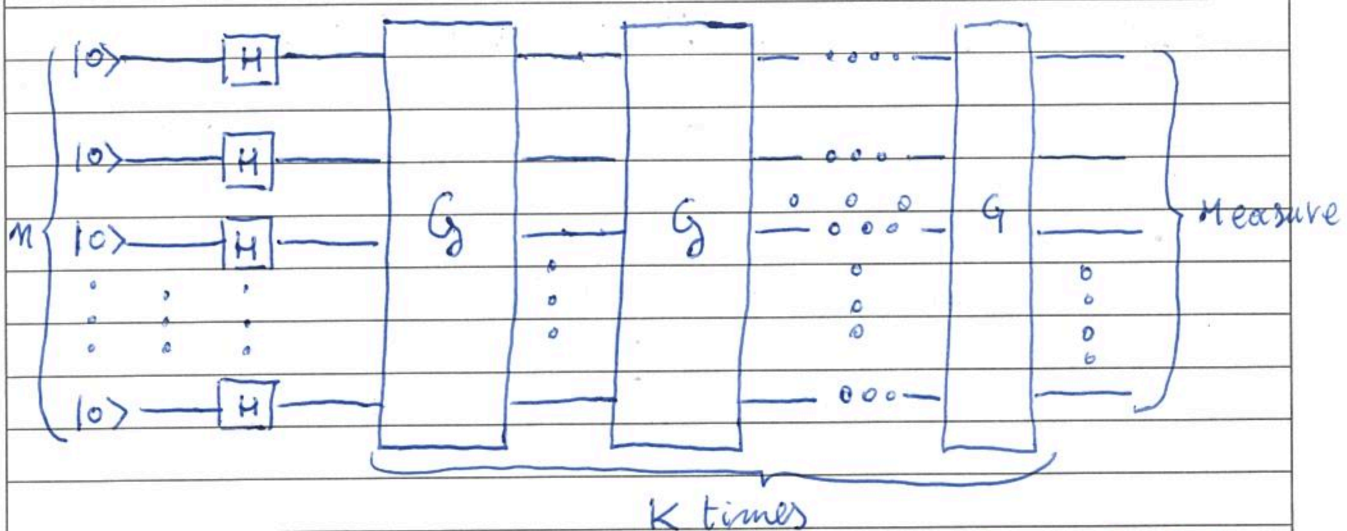
def:
the Grover iterate: $\boxed{G \equiv H^{\otimes n} R_0 H^{\otimes n} O_{x\pm}}$ ← 1 G ↔ 1 Query

where:

- $O_{x\pm}: |i\rangle \to (-)^{x_i}|i\rangle$ (the standard phase query)

- R_0: $\begin{cases} R_0|0^n\rangle \to +|0^n\rangle \\ R_0|i\rangle \to -|i\rangle \ \forall i \neq 0^n \end{cases}$ → can be implemented with $O(n)$ elementary gates

Grover algorithm: circuit



→ talk about it later

Why it work:

after $H^{\otimes n}$ the state is $|U\rangle = \frac{1}{\sqrt{N}} \sum_i |i\rangle$ ⇒ uniform superposition

⇒ $\boxed{\text{we need to move amplitudes to states } |j\rangle \text{ s.t. } x_j = 1}$

⇒ define 2 states: "good", $|G\rangle$ and "Bad", $|B\rangle$.

$$|G\rangle = \frac{1}{\sqrt{t}} \sum_{i \text{ s.t. } x_i = 1} |i\rangle \qquad |B\rangle = \frac{1}{\sqrt{N-t}} \sum_{i \text{ s.t. } x_i = 0} |i\rangle$$

$$\Rightarrow |U\rangle = \frac{1}{\sqrt{N}} \left(\sum_{i : x_i = 1} |i\rangle + \sum_{i : x_i = 0} |i\rangle \right) = \frac{\sqrt{t}}{\sqrt{N}} |G\rangle + \frac{\sqrt{N-t}}{\sqrt{N}} |B\rangle$$

notice $\left(\frac{\sqrt{t}}{\sqrt{N}}\right)^2 + \left(\frac{\sqrt{N-t}}{\sqrt{N}}\right)^2 = 1$ ⇒ $\boxed{\begin{array}{l} |U\rangle = \sin\theta\, |G\rangle + \cos\theta\, |B\rangle \\ \theta = \arcsin\left(\sqrt{\frac{t}{N}}\right) \end{array}}$

⇒ $\boxed{\text{2D space, we want to move towards } \theta = \frac{\pi}{2}}$

- **Next ingredient**: G as a product of two reflections

reflections through subspace V

An operator O is a reflection through a subspace V if

$$\boxed{\begin{array}{l} O|v\rangle = v \quad \forall v \in V \\ O|w\rangle = -|w\rangle \ \forall w : \langle w|v\rangle = 0 \end{array}} \Rightarrow \boxed{O = 2P_V - \mathbb{1}}$$

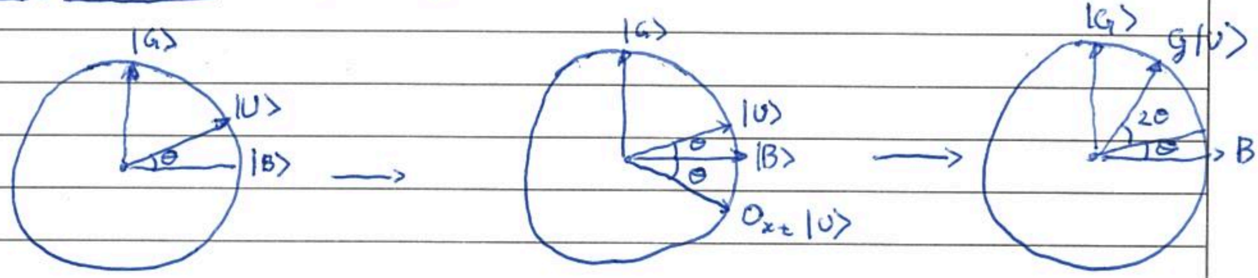
projector on V

⇒ • $O_{x\pm}$ is a reflection through $|B\rangle$

• $H^{\otimes n} R_0 H^{\otimes n} = H^{\otimes n}(2|0^n\rangle\langle 0^n| - \mathbb{1}) H^{\otimes n} = 2H^{\otimes n}|0^n\rangle\langle 0^n|H^{\otimes n} - \mathbb{1} =$

$= 2|U\rangle\langle U| - \mathbb{1}$ ⇒ reflection through $|U\rangle$

the $\mathcal{G}$-action: (graphical)



$\Rightarrow$ $\boxed{\mathcal{G}|U\rangle \text{ has moved towards } |G\rangle}$

$\Rightarrow$ after k-steps: $\boxed{|\psi_k\rangle = \sin\left((2k+1)\theta\right)|G\rangle + \cos\left((2k+1)\theta\right)|B\rangle}$

$\Rightarrow$ the probability of success is: $\boxed{P_k = \sin^2\left((2k+1)\theta\right)}$

- what is the best k?

we want $(2\tilde{k}+1)\theta = \frac{\pi}{2} \Rightarrow 2\tilde{k}+1 = \frac{\pi}{2\theta}$

$\Rightarrow \boxed{\tilde{k} \sim \frac{\pi}{4\theta} - \frac{1}{2}}$ ← notice that $\tilde{k}$ must be an integer ~~so~~ so we cannot really achieve it (depending on θ, but if the number of solutions t is known, something can be done)

$\tilde{k} \sim \frac{\pi}{4\arcsin\left(\sqrt{\frac{t}{N}}\right)} - \frac{1}{2} \Rightarrow \tilde{k} \sim \frac{\pi}{4}\sqrt{\frac{N}{t}}$

$\Rightarrow$ complexity: $O(k) \sim O(\sqrt{N})$ quadratic advantage

Unidade Journeys into theoretical physics 2024
Disciplina 4 lectures on quantum computing: Lecture 4, part 1
Docente
Aluno
RA Período 22÷26/07/2024 Data / /

Lecture 4: random walk algorithms

Setup: - We have a graph $G(E, V)$

Edges; vertices → N in total

- $\varepsilon = \frac{t}{N}$ - fraction is marked

Goal: Find $1\ v \in V$ which is marked

→ clearly very much in the spirit of **Grover**

→ We want to solve it in a random walk fashion:

1) start from a given vertex y
2) check if marked
3) if not, move to a neighbor
4) keep repeating 2) + 3) until we find it

⇒ convenient in terms of memory:

N vertices ⇒ 2^n bits

y requires $O(n) = O(\log N)$

■ A Mathematical tool: the normalized adjacency matrix P

- restrict to $G(E,V)$ being:
  - connected: $\forall (v_1, v_2)$ you can connect them
  - d-regular: $\forall\ v_i$ has exactly d-neighbors
  - no bipartite ~~self-loops: no Edges like (v_i, v_i)~~

⇒ we define P to ~~ve~~ be:

$$\boxed{\begin{aligned} P:\ N\times N \text{ matrix}: \quad & P_{v_1 v_2} = 0 \text{ if } \nexists\, e\in E \text{ connecting } (v_1, v_2) \\ & P_{v_1 v_2} = \frac{1}{d} \text{ if } \exists \text{ '' '' '' '' '' } \end{aligned}}$$

⇒ ~~Pav~~ let's take $\hat{v} = \begin{pmatrix} 0\\0\\0\\1\\0\\0 \end{pmatrix}$ N-dimensional, normalized, vector

⇒ $P\hat{v} = \left.\begin{pmatrix} 1/d\\0\\0\\1/d\\0\\0 \end{pmatrix}\right\}$ in d places ⇒ it can be used to represent a step of the random walk

⇒ I will be with $p = \frac{1}{d}$ in one of the neighbours

■ Properties of P: (mostly stated without proof)

let us define $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N$ ← Eigenvalues

$v_1, v_2, \ldots v_N$ ← Eigenvectors

⇒ it's possible to show:

$\lambda_1 = 1$ → dominant pair and also $\boxed{\lambda_N > -1}$

$v_1 = \frac{1}{N}$ ← uniform distribution

⇒ we can show (Exercise)

$$\boxed{\begin{aligned} & P^K v \sim v_1 \\ & \text{if } K \sim \frac{1}{|\lambda_1 - \lambda_2|} \equiv \frac{1}{\underset{\text{spectral gap}}{\delta}} \end{aligned}}$$

$\Rightarrow$ We are now ready for the $\boxed{\text{classical random walk algorithm}}$

1) Prepare $\hat{v}$ inicial

2) evolve $K \sim \frac{1}{\delta}$ times $\Rightarrow \hat{v_f} = P^K v_i \sim U$

3) extract 1 vertex from $U \Rightarrow y$ is marked with $p = \varepsilon$

4) repeat 2) and 3) $\sim \frac{1}{\varepsilon}$ Times

$\Rightarrow$ Total complexity:

$$\mathcal{C} \simeq S + \frac{1}{\epsilon}\left(C + \frac{1}{\delta} U\right)$$

S: setting the initial state (queries to create v)

C: cost to check if a vertex is marked

U: cost to update: $v \sim Pv$

notice that this query complexity is in the spirit of when we have a small but fast RAM, and a large but slow HD: the real consuming operations are the calls to the HD

Quantum algorithm : GROVER + GRAPH twist

in the quantum algorithm we need 2 registers: $\underbrace{|x\rangle |y\rangle}^{\text{Edge}}$
current vertex   past vertex

- define: $\boxed{|p_x\rangle = \sum_y \sqrt{P_{xy}}\, |y\rangle}$ → uniform superposition of neighbors

⇒ like in GROVER define "good" and "bad" states

- M: {marked vertices}

$$|G\rangle = \frac{1}{\sqrt{M}} \sum_{x \in M} |x\rangle |p_x\rangle \qquad \cdot\ |B\rangle = \frac{1}{\sqrt{N-M}} \sum_{x \notin M} |x\rangle |p_x\rangle$$

⇒ define $\epsilon = |M|/N$ $\theta = \arcsin(\sqrt{\epsilon})$

and the uniform superposition (Easy to make with Hadamard)

$$\boxed{|U\rangle = \frac{1}{\sqrt{N}} \sum_x |x\rangle |p_x\rangle = \sin(\theta) |G\rangle + \cos(\theta) |B\rangle}$$

⇒ the quantum Algorithm, GROVER-like :

1) setup $|U\rangle$

   from GROVER

2) repeat the following $O(1/\sqrt{\epsilon})$ times

   2.a) Reflect through $|B\rangle$
   2.b) Reflect through $|U\rangle$

   → this requires R.W. technology

3) Measure and check.

   for the usual reasons this is going to find a marked vertex with high probability.

Unidade Journey into theoretical physics 2024
Disciplina 4 lectures on quantum computing: Lecture 4, part 2
Docente
Aluno
RA Período 22 ÷ 26/07/2024 Data / /

How we can implement (2.a) and (2.b)?

→ (2.a) is ~~easy~~ easy: we make a ~~sto~~ "phase" Query-like:

$$\text{if } |x\rangle \text{ marked} \Rightarrow -|x\rangle$$
$$\text{if } |x\rangle \text{ not marked} \Rightarrow +|x\rangle$$

(2.b) this is much harder to implement and where R.W. enter:

- General idea: classical intuition:

we have P and $P^k v \to \begin{pmatrix} 1/N \\ 1/N \\ \vdots \\ 1/N \end{pmatrix}$ uniform

⇒ we need, probably, a "quantum encoding of P"

and then some repeated action of P will give a projection on U

Let's see how we can do this in practice:

- define:

$\mathcal{A} = \text{span}\{|x\rangle|p_x\rangle\} \Rightarrow \text{ref}(\mathcal{A})$: reflection through $\mathcal{A}$

$\mathcal{B} = \text{span}\{|p_y\rangle|y\rangle\} \Rightarrow \text{ref}(\mathcal{B})$: reflection through B

$\boxed{W(P) = \text{ref}(B)\,\text{ref}(\mathcal{A})}$ this is going to be our main actor (i.e. the "quantum encoding of P")

- How to implement this?

consider the following operation (possibly controlled)

(1) $|x\rangle|0\rangle \longrightarrow |x\rangle|p_x\rangle$ } this is like making a step of
(2) $|0\rangle|y\rangle \longrightarrow |p_y\rangle|y\rangle$ } random walk

$\Rightarrow$ ref(A): $(1)^{-1} \longrightarrow (-1) \xrightarrow{\text{if 2nd reg} \neq 0} (1)$

ref(B): $(2)^{-1} \longrightarrow (-1) \xrightarrow{\text{if 1st register} \neq 0} (2)$

$\Rightarrow$ $\boxed{\text{we can implement the action of } W(P) \text{ via 4 Random Walk steps.}}$

Eigen spectrum of $W(P)$: related to the $\lambda_j \Rightarrow$ call them $\tilde{\lambda}_j$

let $|\lambda_j| = \cos\theta_j \quad \theta_j \in [0, \frac{\pi}{2}]$

$\Rightarrow \boxed{\tilde{\lambda}_j = e^{\pm 2i\theta_j}}$ and $|U\rangle$ is eigenvector with $\tilde{\lambda}_1 = 1 \Rightarrow \theta_1 = 0$

$\Rightarrow \tilde{\lambda}_j = e^{\pm 2i\theta_j}$ $\boxed{\theta_j \geq \sqrt{2\delta}}$
all others

taylor

$1 - \delta \geq |\lambda_j| = \cos\theta_j \geq 1 - \frac{\theta_j^2}{2}$

$\Rightarrow$ $\boxed{\text{we finally obtain that the reflection through } |U\rangle \text{ can be done via a reflection through the eigenvalue-1 space of } W(P)}$

How can we do this?

||

$\boxed{\text{phase estimation with significance } \frac{\sqrt{\delta}}{2}}$ $\Rightarrow$ requires $O\left(\frac{1}{\sqrt{\delta}}\right)$ applications of $W(P)$

" $O\left(\log\left(\frac{1}{\delta}\right)\right)$ auxiliary qubits

$\Rightarrow$ Schematically:

$$R(P): |w\rangle|0\rangle \xrightarrow{P.E.} |w\rangle|\overset{\text{appro}}{\tilde{\Theta}_j}\rangle \longrightarrow (-1)^{[\theta_j \neq 0]} |w\rangle|\tilde{\theta}_j\rangle \xrightarrow{P.E.^{-1}} |w\rangle|0\rangle$$

the wanted reflection

and this conclude the implementation of (2.b.)

$\Rightarrow$ we can now count the $\boxed{\text{COMPLEXITY of the algorithm}}$

- Setup cost: cost of building $|U\rangle \Rightarrow S$
- checking cost: cost of ~~ehee~~ the map $|x\rangle|y\rangle \longrightarrow m_x |x\rangle|y\rangle \Rightarrow C$

  -1 marked
  +1 not

- Update cost: cost of one R.W. step: cost of $W(P) \Rightarrow U$

$\Rightarrow$ total complexity

$$\boxed{S + \underbrace{\frac{1}{\sqrt{\varepsilon}}}_{\text{grover}}\left(C + \underbrace{\frac{1}{\sqrt{\delta}}}_{\text{phase estimation}} U\right)}$$

- compared to the classical case: $\left.\begin{array}{l}\varepsilon \longrightarrow \sqrt{\varepsilon} \\ \delta \longrightarrow \sqrt{\delta}\end{array}\right] \longrightarrow$ advantage.

■ Application: the collision problem

Input: $x = x_0, x_1, \ldots, x_{n-1}$ $\quad x_i \in \mathbb{N}$

Goal: Find i, j $i \neq j$ s.t. $x_i = x_j$, ~~see~~ if existent. $\longrightarrow \boxed{\text{classical } O(n)}$

This problem can be solved using a Random walk on a $\boxed{\text{Johnson graph}}$

■ the Johnson graph : $\boxed{J(n, r)}$

- to consider $r < n$

⇒ vertices: sets $R \subseteq \{0, \_, n-1\}$ with r elements.

⇒ $N = \binom{n}{r}$

⇒ Edges : R, R' connected iff: R' can be obtained by removing 1 element from R and adding 1 element.

→ $\boxed{J(n,r) \text{ is } r(n-r)\text{-regular}}$

property : spectral gap is $\boxed{\delta = \frac{n}{r(n-r)} \approx \frac{1}{r} \quad n \gg r}$

• For our algorithm :

- our vertex contains (R, x_R) the elements x_i with $i \in R$

- what about ε? ⇒ worst case scenario: 1 collision (i,j)

→ $\boxed{\varepsilon = \frac{r}{n} \cdot \frac{r-1}{n-1} \approx \left(\frac{r}{n}\right)^2}$

⇒ algorithm cost (in terms of queries to oracle x)

S: $r+1$ ($|U\rangle$ is a uniform superposition and $|R\rangle|R'\rangle$ contain $r+1$ elements)

U: $O(1)$ (I need to query the new added element)

C: $=0$ (no query required)

$$\Rightarrow S + \frac{1}{\sqrt{\varepsilon}}\left(C + \frac{1}{\sqrt{\delta}} U\right) = O\left(r + \frac{n}{\sqrt{r}}\right) \sim O\left(n^{2/3}\right)$$

better than classical

→ for $r = n^{2/3}$