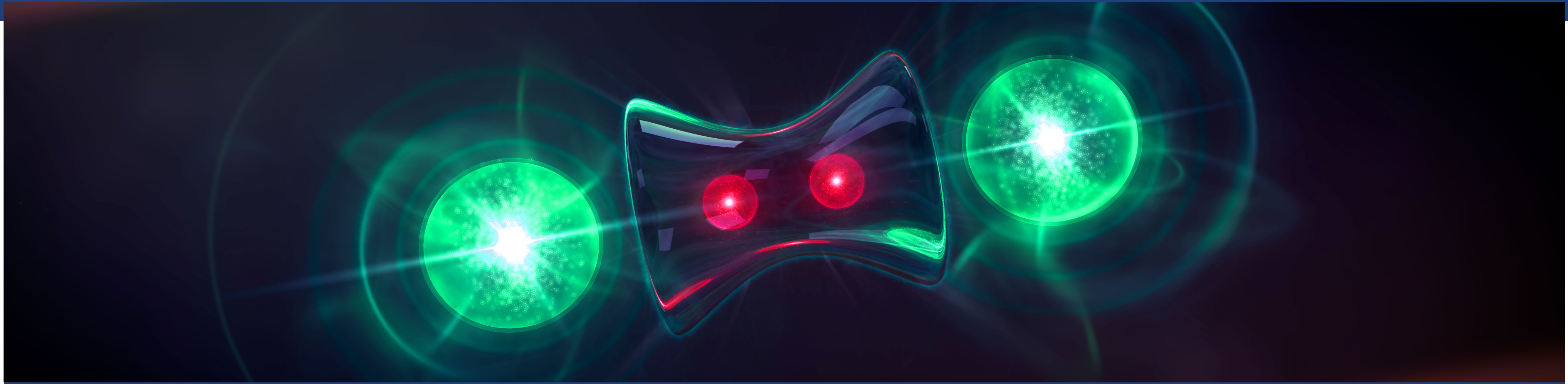


The Grover's Algorithm

Camilo Zapata
Xiao Yang

The problem

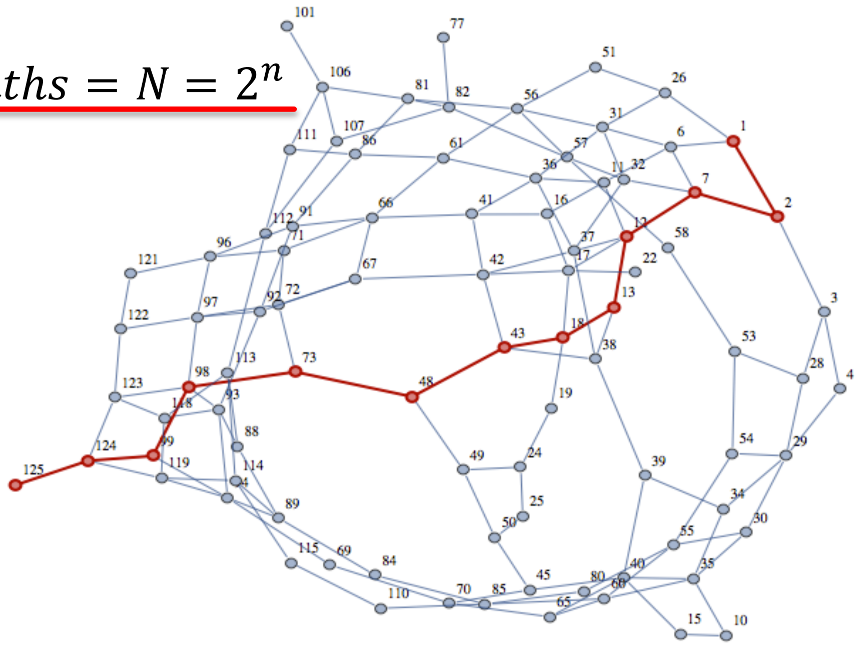
Find a solution to a certain problem in a unordered list of possibilities.

$Entries = N = 2^n$

Person	Height	Age	Weight	IQ
John	173	25	200	95
Peter	175	26	185	75
Greg	195	32	191	65
James	165	28	160	150
Matthew	152	15	140	135
Peter	145	12	130	100

Find element on a database.

$Paths = N = 2^n$



Find shortest path between two cities

Classically…

- In the classical case it is obvious that the performance of the search algorithm is **O(N)**.
- The expected running time is **~N/2**.
- As the data base is unstructured the element that you are looking for could be on the last (N^{th}) entry.
- You would have to perform at most N steps to get the result.

How does the Grover's Algorithm perform?

- Given a list with N elements, the performance of the Grover's Algorithm will be:

$$O(\sqrt{N})$$

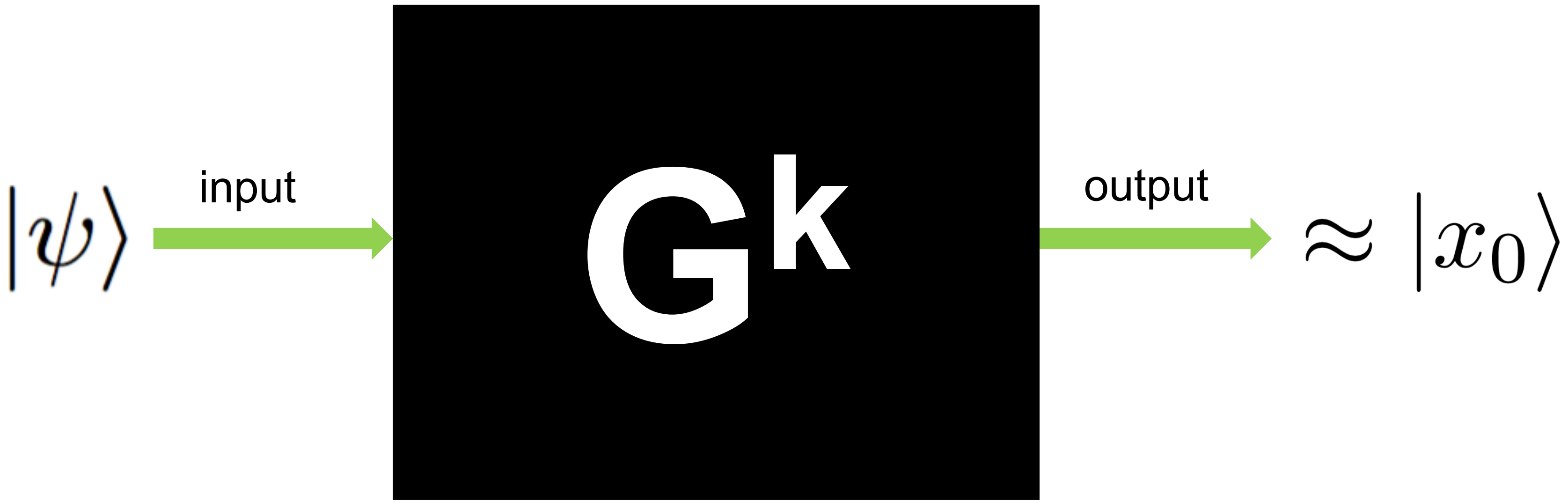
- Meaning that for a list of **10^6 elements**, you would have to iterate the search a number of the order of **1000**.

- As for every quantum algorithm the result will be accurate with a **probability close to 1**.

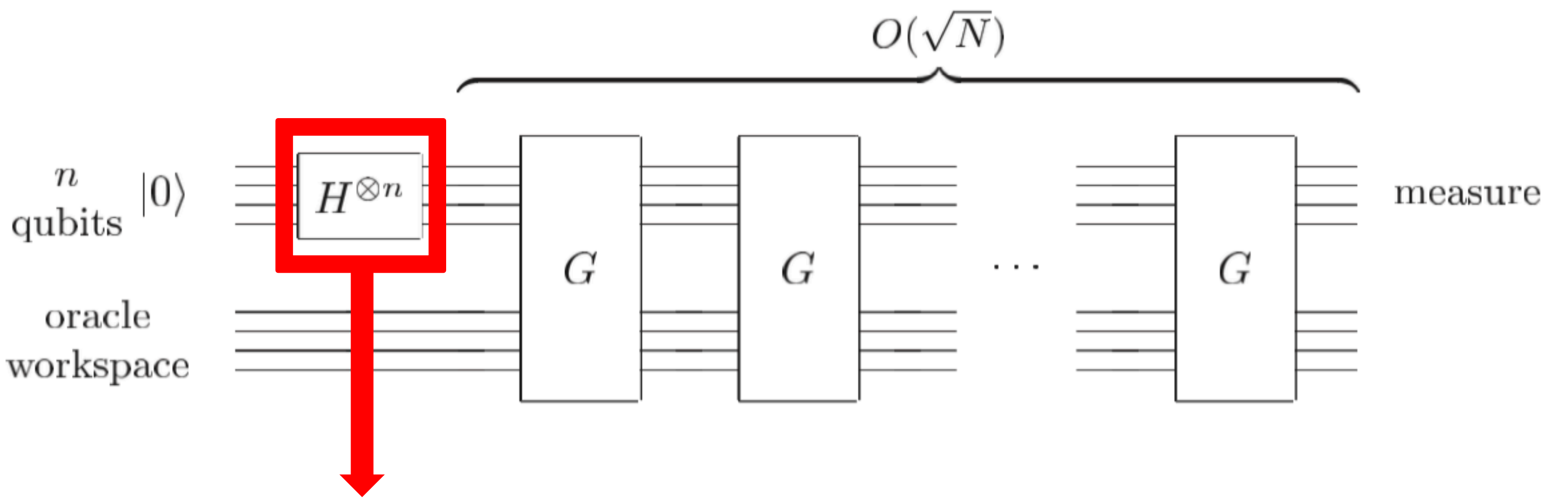
The algorithm

Let the search space be $S = \{|0\rangle, \ldots, |N\rangle\}$ and $|x_0\rangle \in S$ be the only solution for the search.

Let $|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$



Breaking it down



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

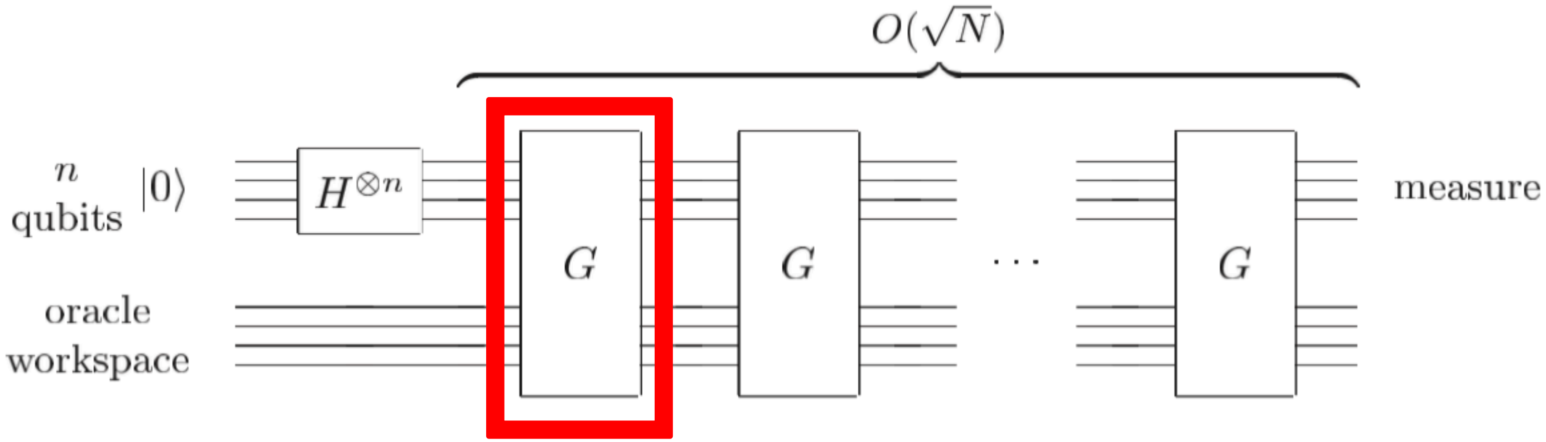
$$H(|0\rangle) = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$

$$H^{\otimes n}|0\rangle^{\otimes n} = |\psi\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle$$

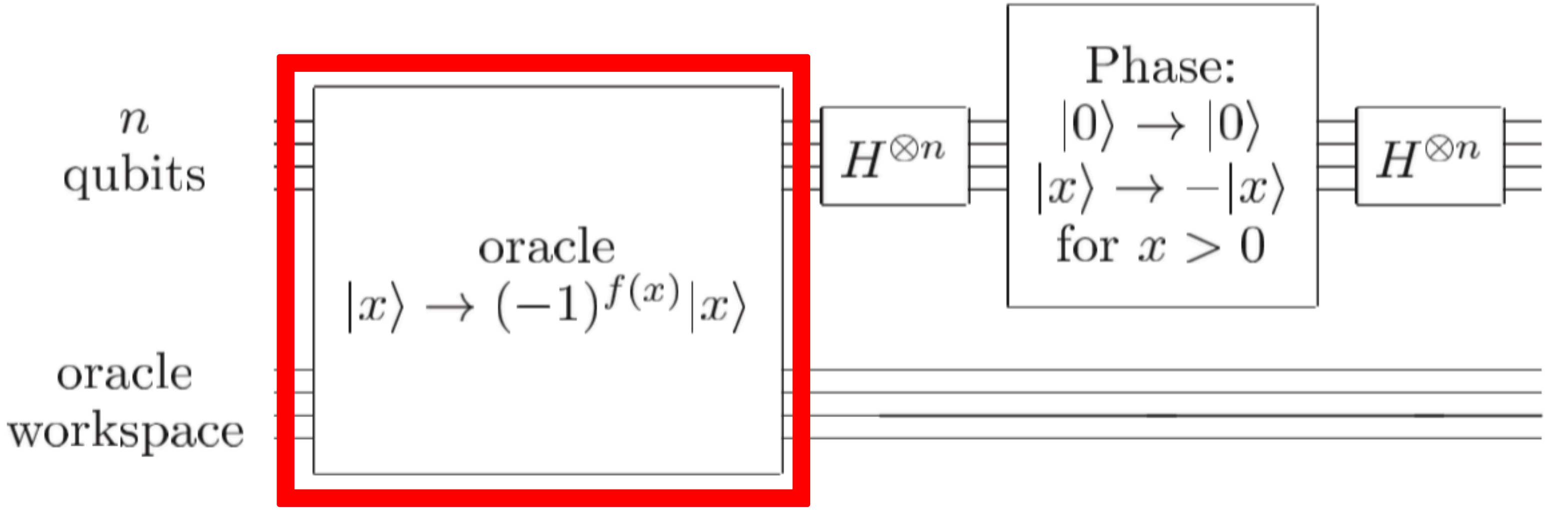
In the case where $n = 2, \quad N = 2^n = 4$

$$|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

Breaking it down



Grover's operator: step 1



- **The oracle:**

If the solution for search is x_0, then:

$$f(x) = \begin{cases} 1 & \text{if } x = x_0 \\ 0 & \text{otherwise} \end{cases}, \qquad |x\rangle \xrightarrow{O} (-1)^{f(x)}|x\rangle$$

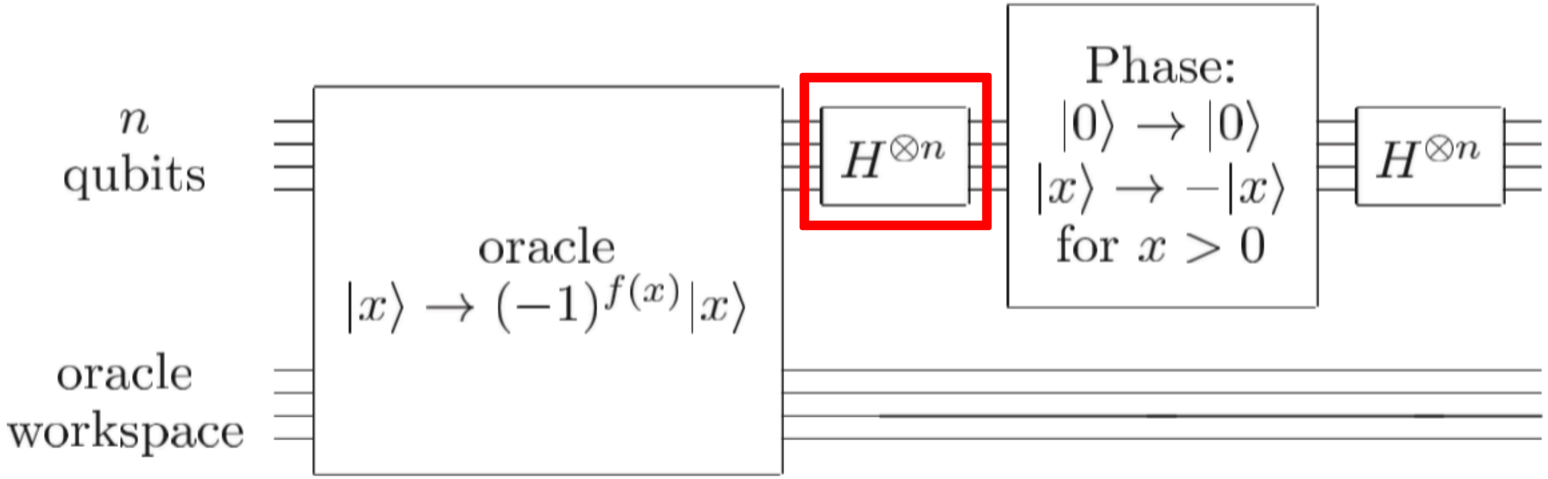
Example N=4

Solution $x_0 = 2$

$$|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

$$\xrightarrow{O} \quad \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle + |11\rangle)$$

Grover's operator: step 2



Example N=4

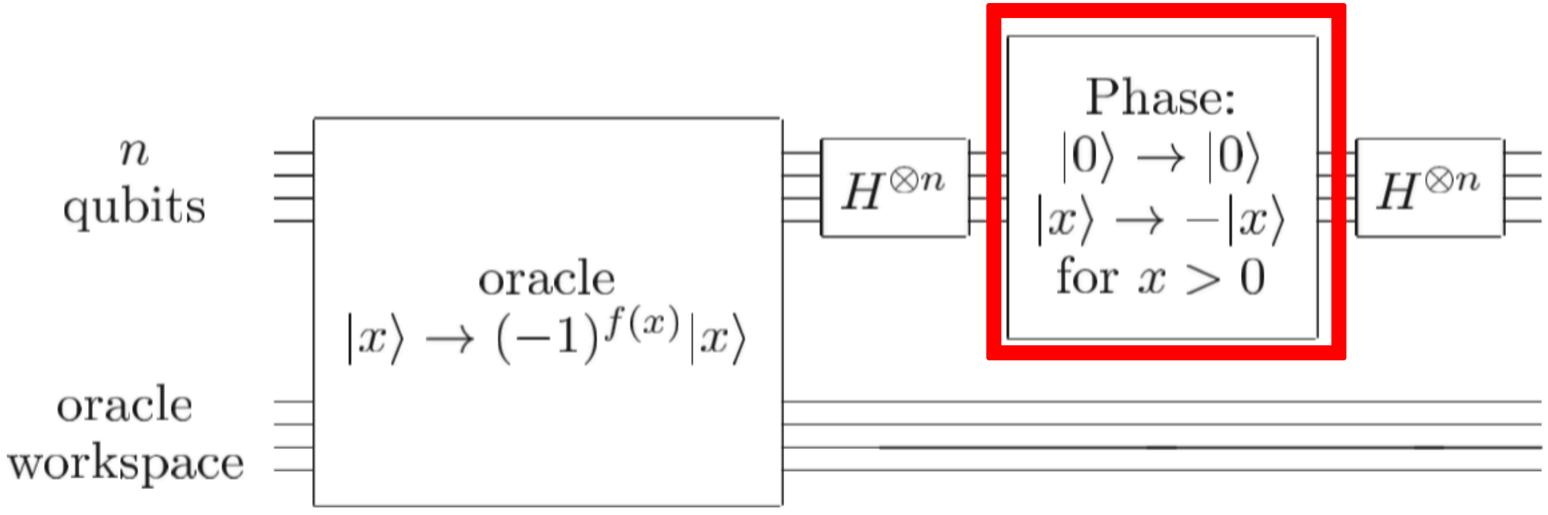
Solution $x_0 = 2$

$$|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

$$\xrightarrow{O} \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle + |11\rangle)$$

$$\xrightarrow{H^{\otimes 2}} \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle + |11\rangle)$$

Grover's operator: step 3



Example N=4

Solution $x_0 = 2$

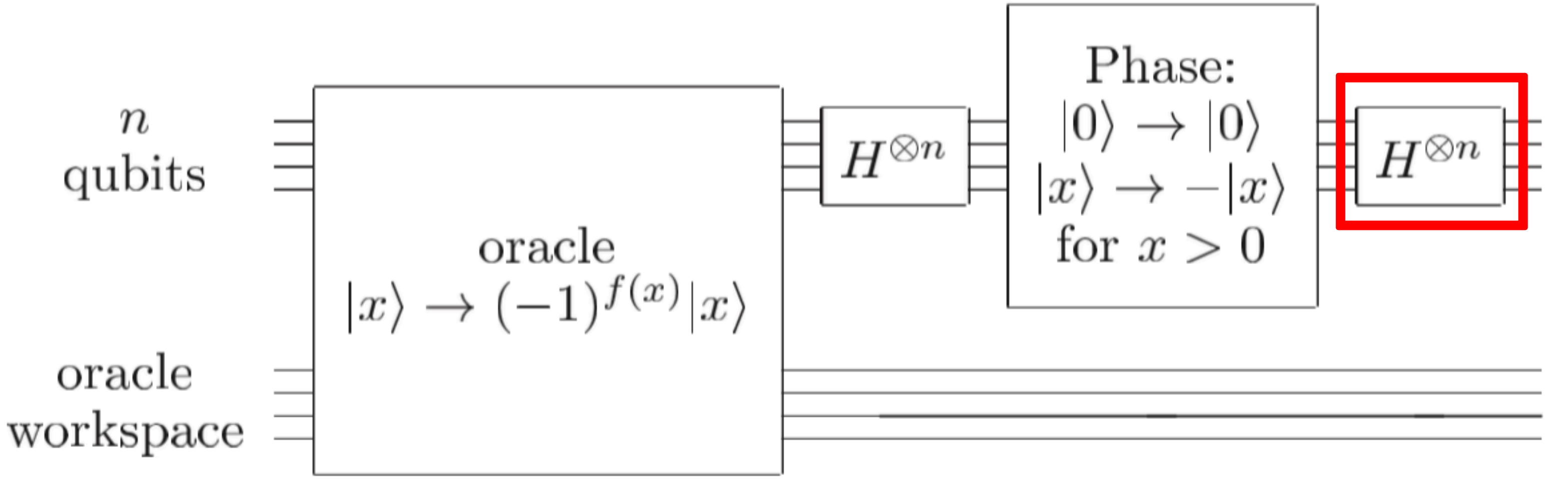
$$|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

$$\xrightarrow{O} \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle + |11\rangle)$$

$$\xrightarrow{H^{\otimes 2}} \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle + |11\rangle)$$

$$\xrightarrow{phase} \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle)$$

Grover's operator: step 4



Example N=4

Solution $x_0 = 2$

$$|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$$

$$\xrightarrow{O} \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle + |11\rangle)$$

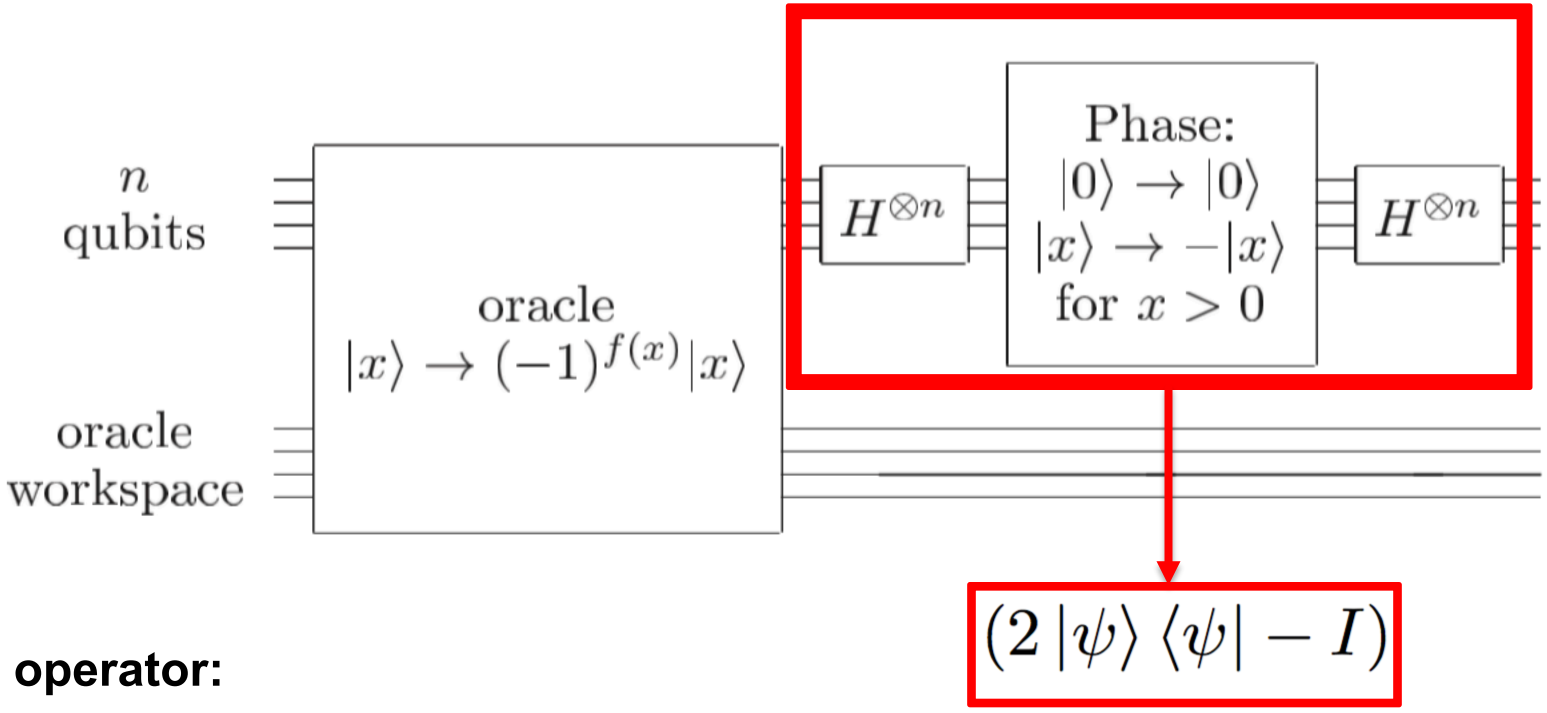
$$\xrightarrow{H^{\otimes 2}} \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle + |11\rangle)$$

$$\xrightarrow{phase} \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle)$$

$$\xrightarrow{H^{\otimes 2}} |10\rangle$$

N=4 is a special case. We get to the desired state with just one Iteration of the Grover's algorithm.

Grover’s operator



- **The Grover’s operator:**

$$G = (2|\psi\rangle\langle\psi| - I)\, O$$

Geometric interpretation

- Define the states: $$|\alpha\rangle \equiv \frac{1}{\sqrt{N-1}} \sum_{x \neq x_0} |x\rangle$$

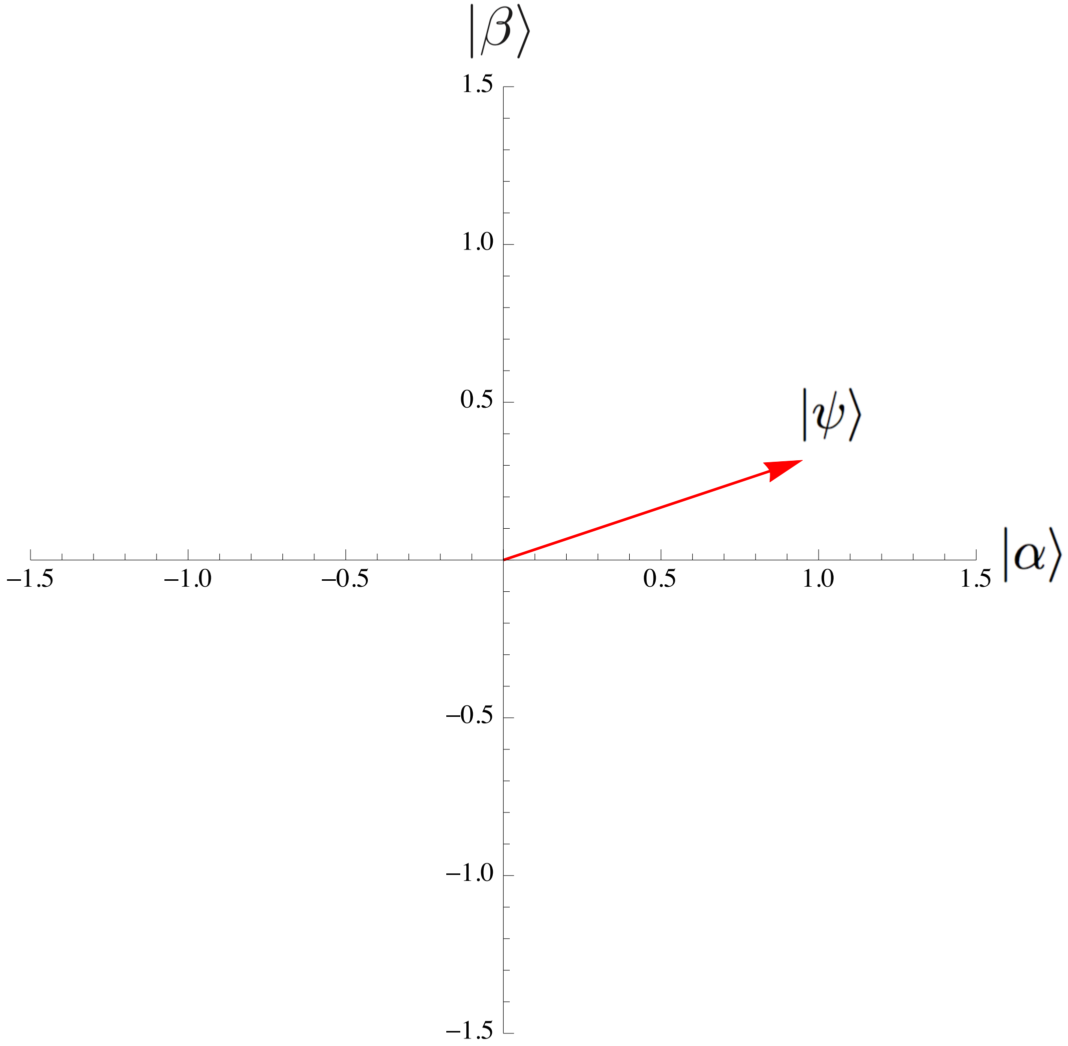
$$|\beta\rangle \equiv |x_0\rangle$$

- It's easy to see that $$|\psi\rangle = \sqrt{\frac{N-1}{N}} |\alpha\rangle + \sqrt{\frac{1}{N}} |\beta\rangle$$

Geometric interpretation

$$G = (2\,|\psi\rangle\,\langle\psi| - I)\,O$$

We start with the state $|\psi\rangle$

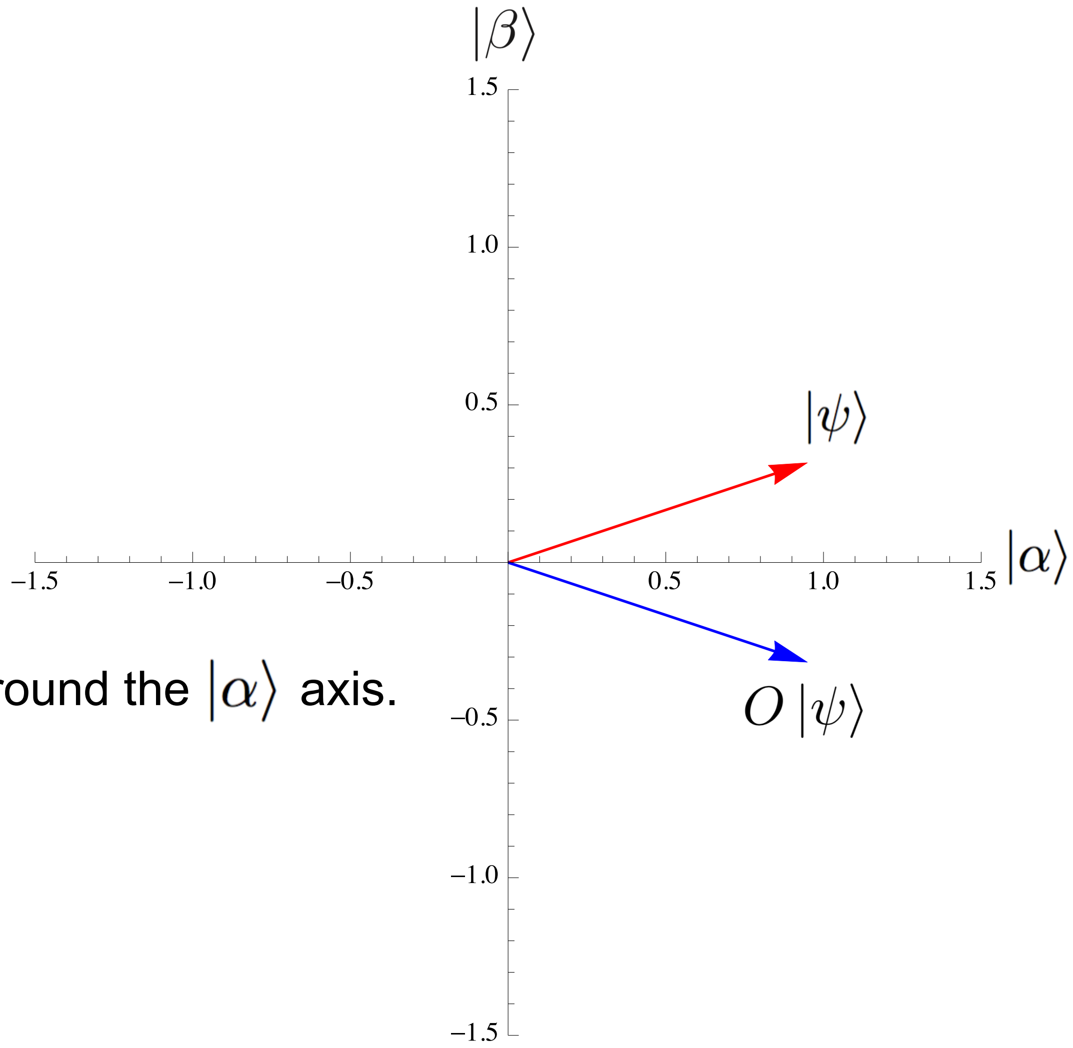


Geometric interpretation

$$G = (2|\psi\rangle\langle\psi| - I)\,O$$

$$O\left(a|\alpha\rangle + b|\beta\rangle\right) = a|\alpha\rangle - b|\beta\rangle$$

A reflection around the $|\alpha\rangle$ axis.

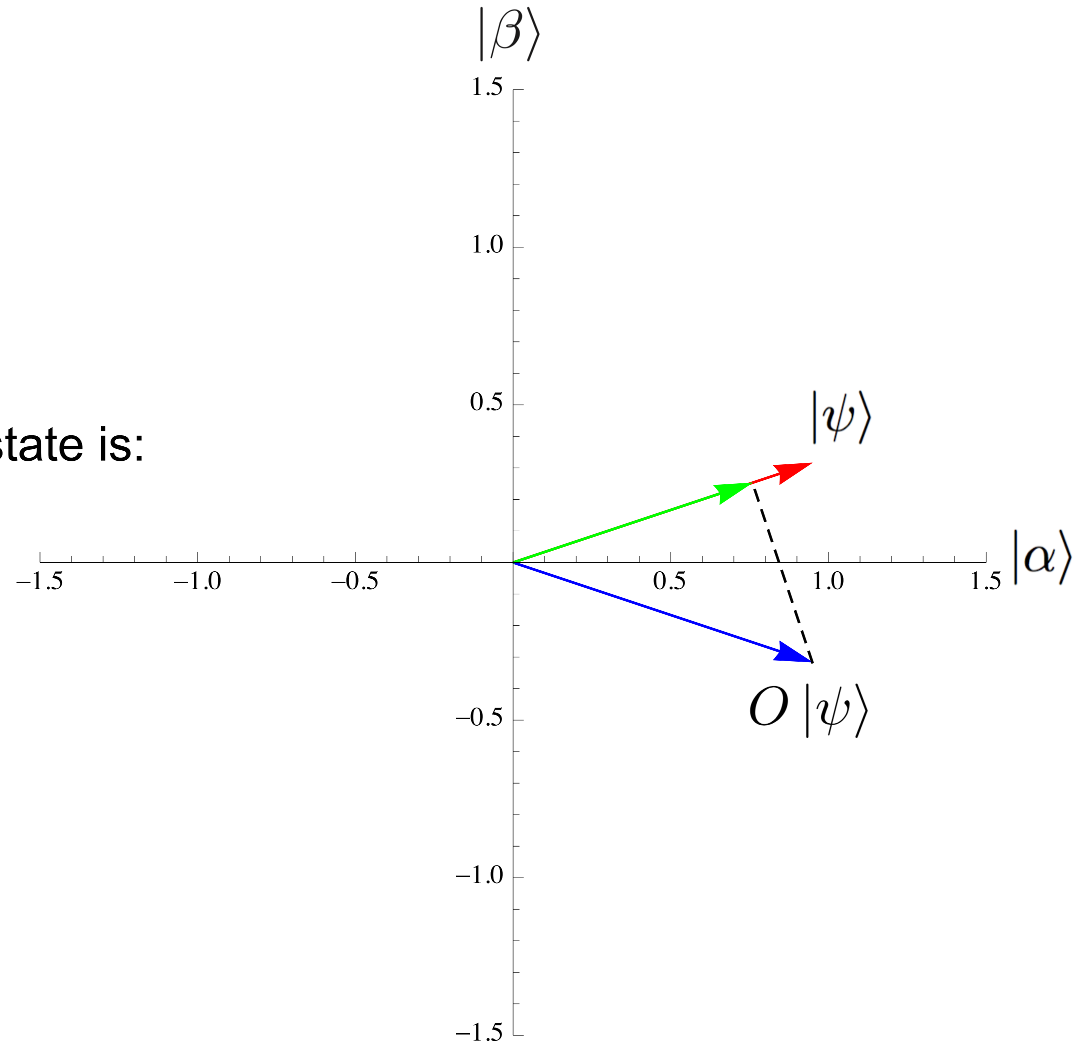


Geometric interpretation

$$G = (2|\psi\rangle\langle\psi| - I)\,O$$

Now, the operator $(2|\psi\rangle\langle\psi| - I)$ applied to a general state is:

$$(2|\psi\rangle\langle\psi| - I)|\phi\rangle = 2\underline{\langle\psi|\phi\rangle|\psi\rangle} - |\phi\rangle$$

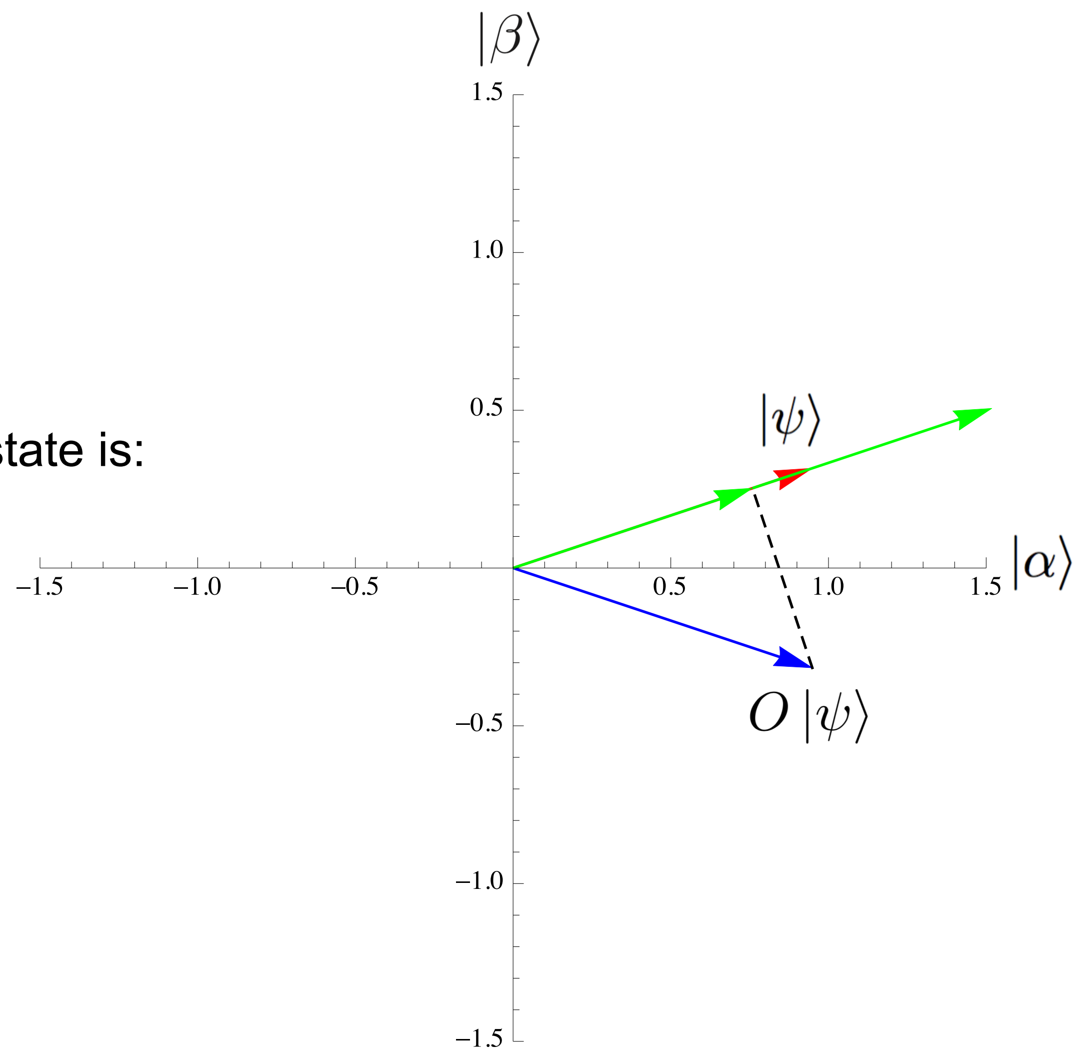


Geometric interpretation

$$G = (2|\psi\rangle\langle\psi| - I)\, O$$

Now, the operator $(2|\psi\rangle\langle\psi| - I)$ applied to a general state is:

$$(2|\psi\rangle\langle\psi| - I)|\phi\rangle = 2\langle\psi|\phi\rangle|\psi\rangle - |\phi\rangle$$

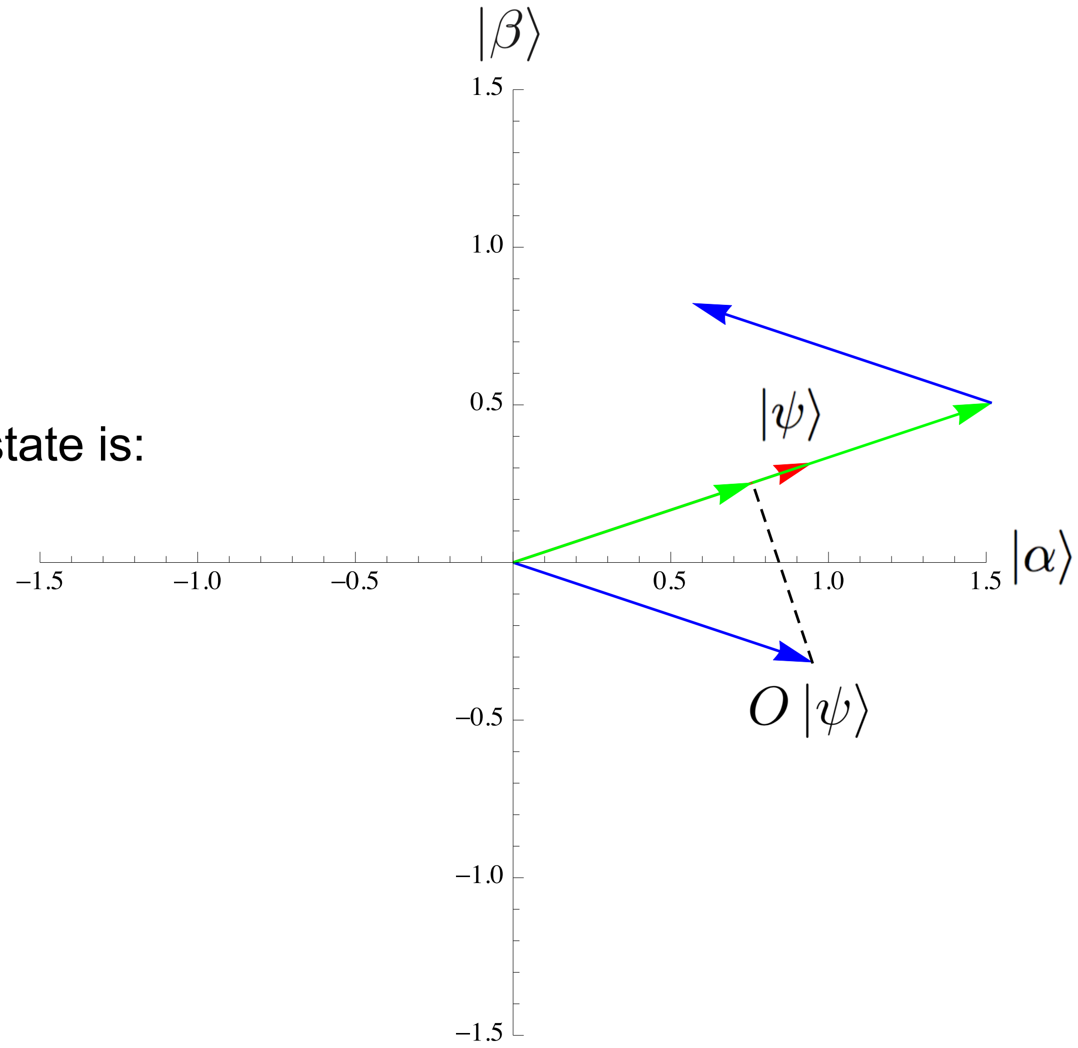


Geometric interpretation

$$G = (2|\psi\rangle\langle\psi| - I)\,O$$

Now, the operator $(2|\psi\rangle\langle\psi| - I)$ applied to a general state is:

$$(2|\psi\rangle\langle\psi| - I)|\phi\rangle = 2\langle\psi|\phi\rangle|\psi\rangle - |\phi\rangle$$



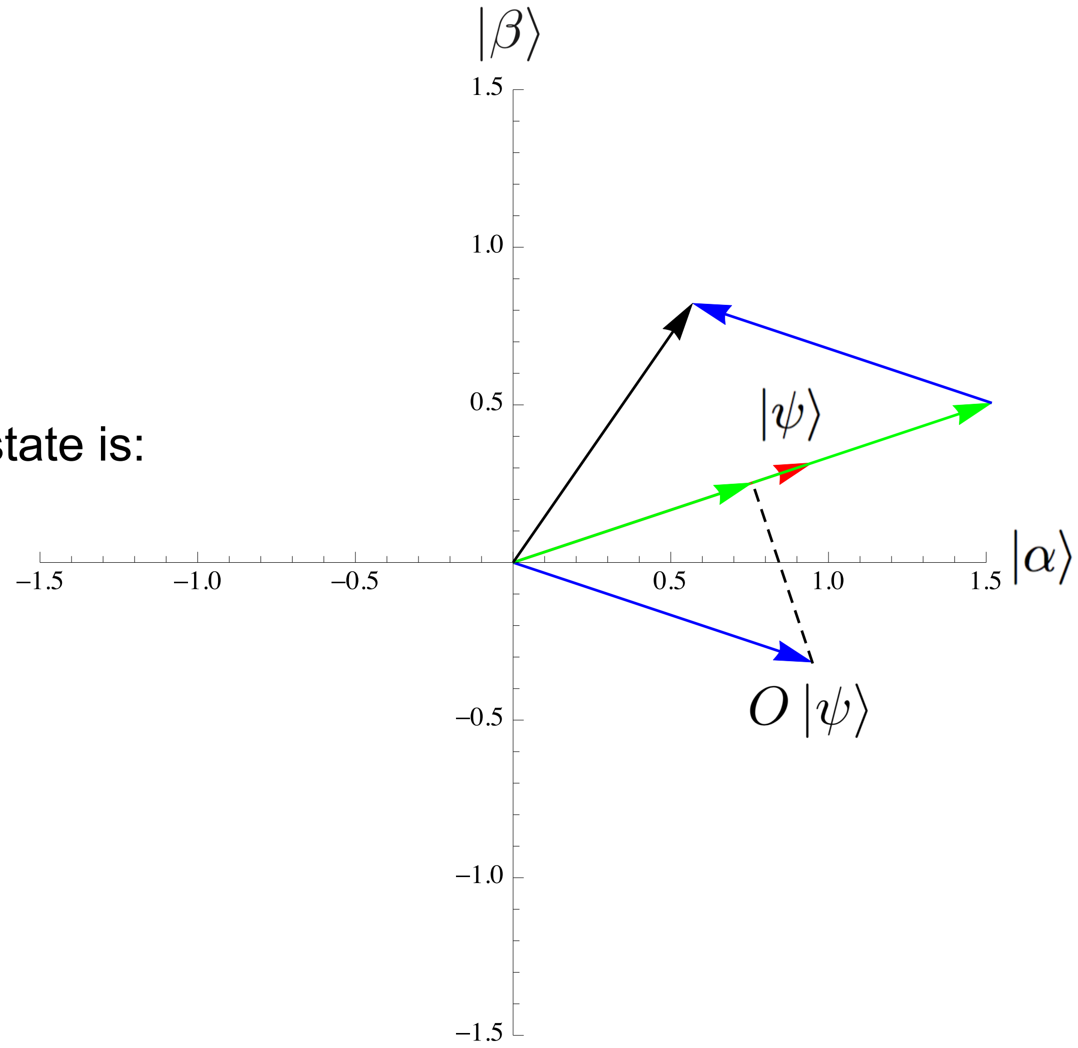
Geometric interpretation

$$G = (2|\psi\rangle\langle\psi| - I)\,O$$

Now, the operator $(2|\psi\rangle\langle\psi| - I)$ applied to a general state is:

$$(2|\psi\rangle\langle\psi| - I)|\phi\rangle = 2\langle\psi|\phi\rangle|\psi\rangle - |\phi\rangle$$

A reflection around the $|\psi\rangle$ state.



Geometric interpretation

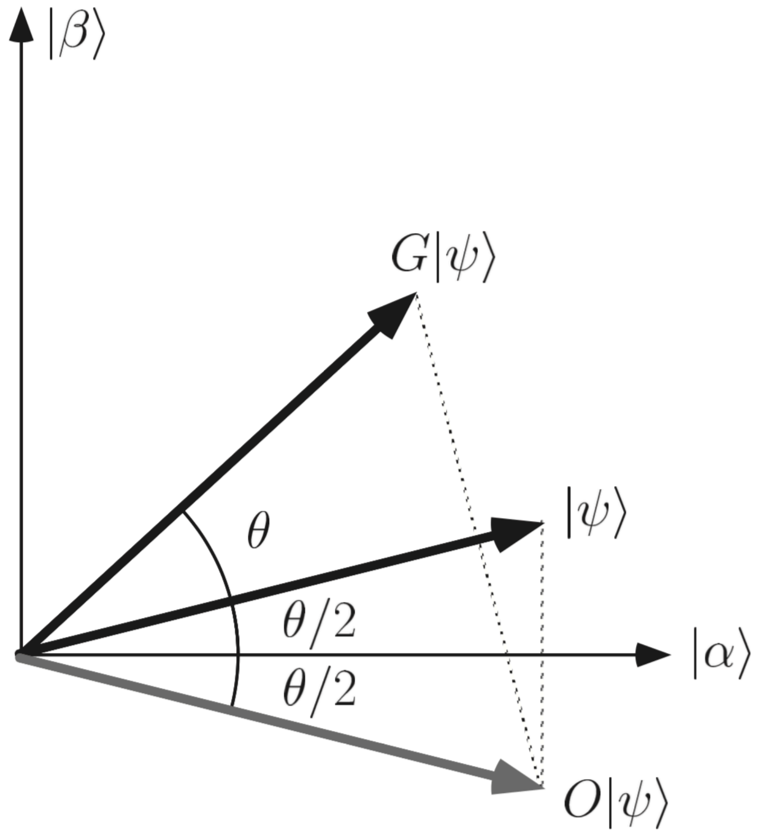
Therefore:

$$G = (2|\psi\rangle\langle\psi| - I)O$$ ← **Is a rotation.**

Then:

$$G|\psi\rangle = \cos\frac{3\theta}{2}|\alpha\rangle + \sin\frac{3\theta}{2}|\beta\rangle \quad , \quad \cos(\theta/2) = \sqrt{\frac{N-1}{N}}$$

$$G^k|\psi\rangle = \cos\left(\frac{2k+1}{2}\theta\right)|\alpha\rangle + \sin\left(\frac{2k+1}{2}\theta\right)|\beta\rangle$$

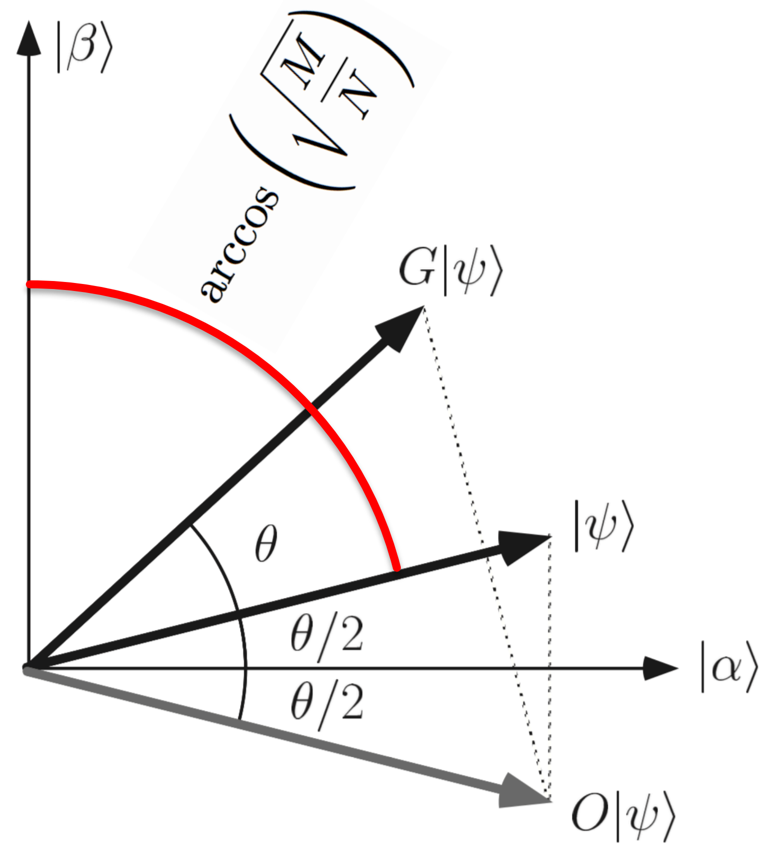


Performance

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}\,|\alpha\rangle + \sqrt{\frac{M}{N}}\,|\beta\rangle$$

The number of times that we have to run the algorithm would be:

$$R = CI\left(\frac{\arccos\left(\sqrt{M/N}\right)}{\theta}\right)$$



Performance

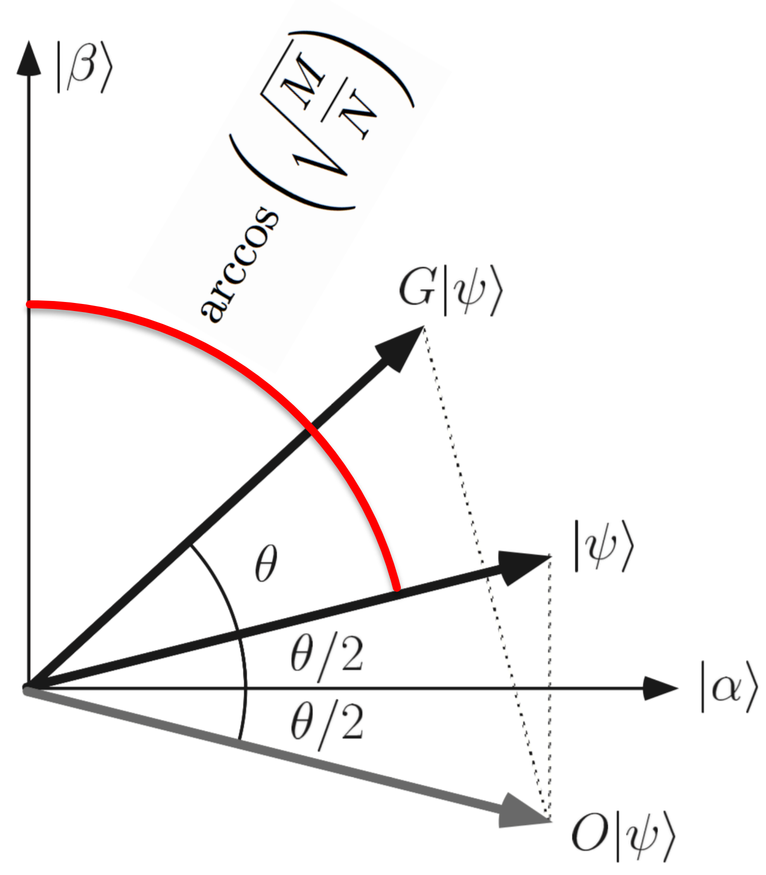
Repeating the algorithm R times rotates $|\psi\rangle$ to within an angle

$\theta/2 \leq \pi/4$ of $|\beta\rangle$.

And making some simple calculations we find an upper bound on R:

$$R \leq \left\lceil \frac{\pi}{4}\sqrt{\frac{N}{M}} \right\rceil$$

So, $R = O(\sqrt{N/M})$

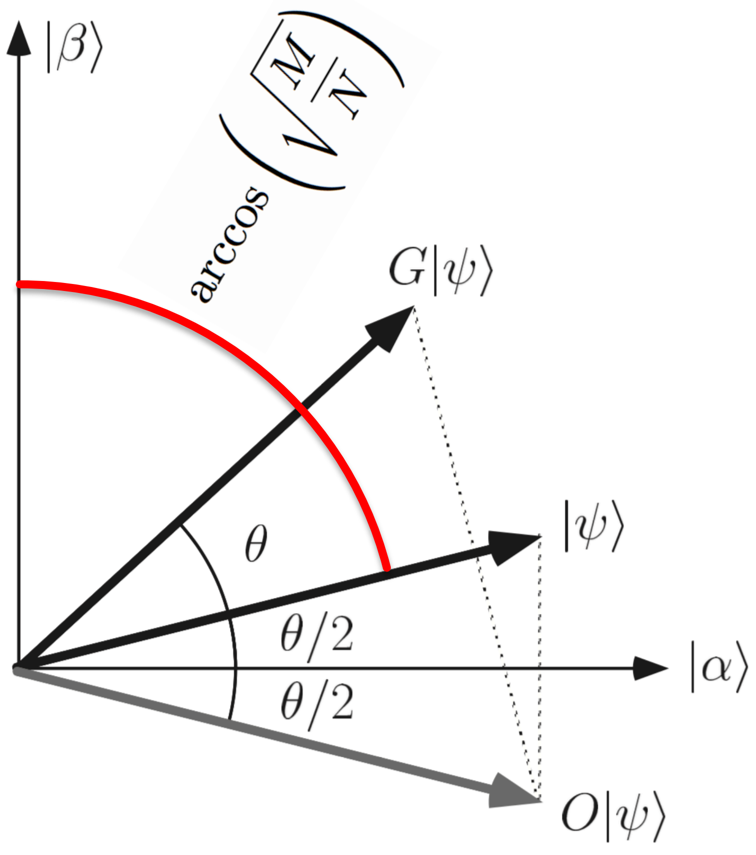


Performance

For M > N/2, we extend the search space by N elements and we get

$$R \leq \left\lceil \frac{\pi}{4}\sqrt{\frac{2N}{M}} \right\rceil$$

So, $R = O(\sqrt{N/M})$



Performance of quantum algorithm

Complexity

- space complexity
- time complexity

$$f(x)=O(g(x)) \quad f(x)\le k|g(x)|$$

$$f(x)=\Omega(g(x)) \quad f(x)\ge k|g(x)|$$

$$f(x)=\Theta(g(x)) \quad n|\mathrm{g}(x)|\le f(x)\le m|g(x)|$$

$$\exists \ n,m,k\in\mathbb{R}^{+}$$

Grover's algorithm

$$R\le\left\lceil\frac{\pi}{4}\sqrt{\frac{N}{M}}\right\rceil=O(\sqrt{N})$$

- In practice, people like to have a polynomial solution rather than exponential ones.

P vs. **NP**

- **P**: Polynomial
  The time-complexity of solution can be done within polynomial time.
- **NP**: Non polynomial

P vs. **NP**

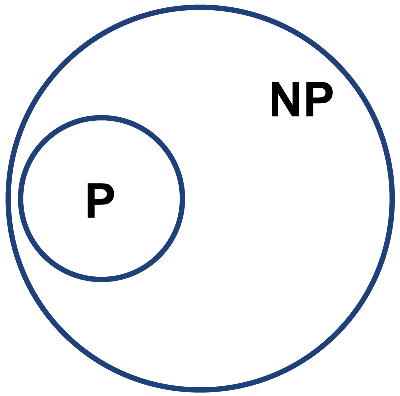
- **P**: Polynomial

  The time-complexity of solution can be done within polynomial time.

- **NP**: Non-deterministic polynomial

  The time-complexity of verifying any “yes”-instances for the solution can be done within polynomial time, but the total solution doesn't have to.
  Apparently, **P** is a subset of **NP**.

  $\mathbf{P} \subseteq \mathbf{NP}$



P vs. **NP**

- **P**: Polynomial

  The time-complexity of solution can be done within polynomial time.

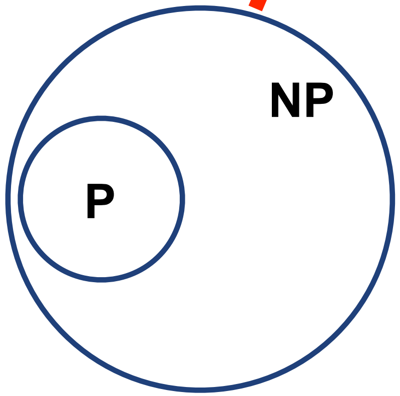
- **NP**: Non-deterministic polynomial

  The time-complexity of verifying any "yes"-instances for the solution can be done within polynomial time, but the total solution doesn't have to.

  Apparently, **P** is a subset of **NP**.

  $\mathbf{P} \subseteq \mathbf{NP}$

Decision problems!!

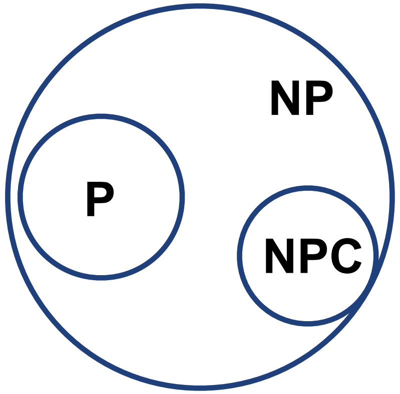


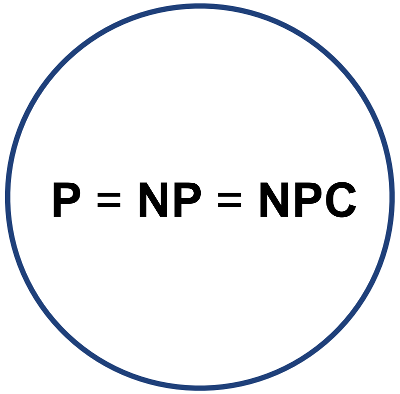
P=NP? conjecture

- **NPC** a subset of **NP**

  Every problem in **NP** is <u>reducible in polynomial time</u> to **NPC**, the hardest problems in **NP**.

  So if one could show that one of the **NPC** problems can be solved in polynomial time, the conjecture is proved. Vice versa, it is disproved.



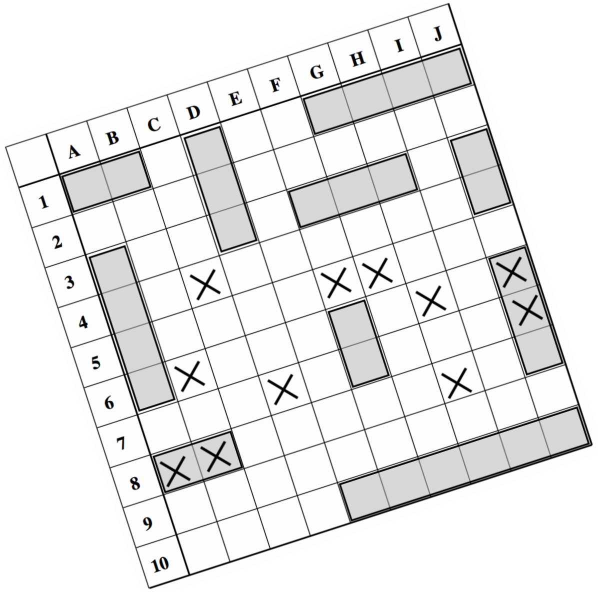


NPC

- Hamiltonian Cycle
- Travelling Salesman
- Subset-partition

NPC

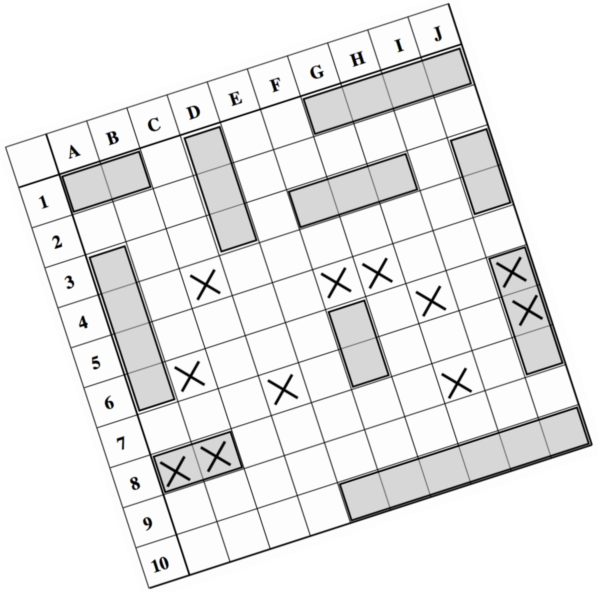
- Hamiltonian Cycle
- Travelling Salesman
- Subset-partition
- Battleship
- Bejeweled



NPC

- Hamiltonian Cycle
- Travelling Salesman
- Subset-partition
- Battleship
- Bejeweled

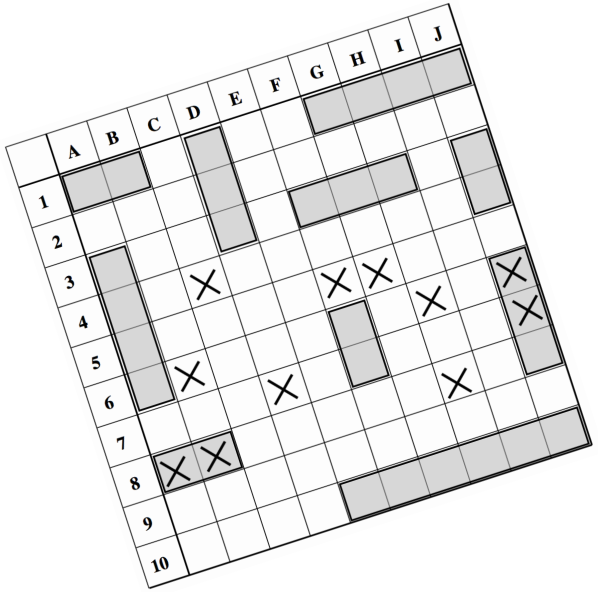
Pokemon???



NPC

- Hamiltonian Cycle
- Travelling Salesman
- Subset-partition
- Battleship
- Bejeweled

Pokemon???



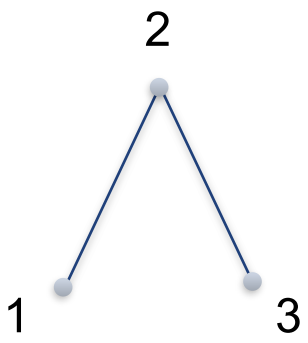
Example 1

- **Hamiltonian Cycle**

Classically, it has complexity of $O\left(p(n)\cdot 2^{n\log n}\right)$

-generate each possible ordering $\left(v_1, v_2, \ldots, v_n\right)$.

-check each ordering for whether it is Hamiltonian cycle.



	1	2	3
1	X	1	0
2	1	X	1
3	0	1	X

123 231

132 312

213 321

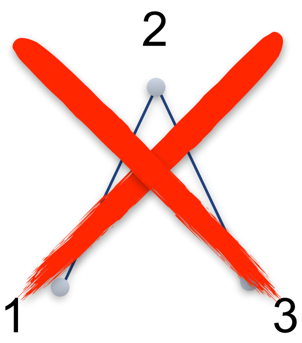
Example 1

- **Hamiltonian Cycle**

Classically, it has complexity of $O\left(p(n)\cdot 2^{n\log n}\right)$

-generate each possible ordering $(v_1, v_2, \ldots, v_n)$.

-check each ordering for whether it is Hamiltonian cycle.



	1	2	3
1	X	1	0
2	1	X	1
3	0	1	X

123 ✗
132 ✗
213 ✗
231 ✗
312 ✗
321 ✗

Example 1

- **Quantum improvement**

  "Hardware" design:
  We need n blocks of vertices index, and each block has m qubits, where $m = \log\lceil n \rceil$. Total size is $S = mn = n\log\lceil n \rceil$ qubits.

  "Software" design:
  Oracle:

$$O|v_1, v_2, \ldots, v_n\rangle = \begin{cases} -|v_1, v_2, \ldots, v_n\rangle & \text{if it is a Hamiltonian cycle} \\ |v_1, v_2, \ldots, v_n\rangle & \text{Otherwise} \end{cases}$$

  Solution ket - qubit-string:

$$|v\rangle := |v_1, v_2, \ldots, v_n\rangle = \bigotimes_{i=1}^{n} |v_i\rangle = \bigotimes_{i=1}^{n} \bigotimes_{j=1}^{m} |\alpha_{ij}\rangle, \quad \alpha_{ij} \in \{0,1\}$$

Index

Block	1	2	…	m
1	|0>	|0>	…	|0>
2	|0>	|0>	…	|1>
…	…	…	…	…
n	|1>	|1>	…	|1>

Example 1

- **Applying quantum search algorithm**
  1. Generate equal superposition state and the oracle qubit.

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{v=0}^{N} |v\rangle, \quad |q\rangle = \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]$$

  2. Apply Grover iteration for $R \approx \lceil \pi\sqrt{N} / 4 \rceil$ times, where $N = 2^S = 2^{mn} = 2^{n\log\lceil n \rceil}$.

$$\left[(2|\psi\rangle\langle\psi| - I)O \right]^R \frac{1}{\sqrt{N}} \sum_{v=0}^{N} |v\rangle|q\rangle \approx |v_0\rangle|q\rangle$$

  3. Measure! It collapses to v_0.

Example 1

- **Applying quantum search algorithm**
  1. Generate equal superposition state and the oracle qubit.

$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{v=0}^{N}|v\rangle, \quad |q\rangle = \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

  2. Apply Grover iteration for $R \approx \lceil \pi\sqrt{N}/4 \rceil$ times, where $N = 2^S = 2^{mn} = 2^{n\log\lceil n\rceil}$.

$$\left[(2|\psi\rangle\langle\psi| - I)O\right]^R \frac{1}{\sqrt{N}}\sum_{v=0}^{N}|v\rangle|q\rangle \approx |v_0\rangle|q\rangle$$

  3. Measure! It collapses to v_0.

Here, we can easily see the complexity is square root of classical algorithm.

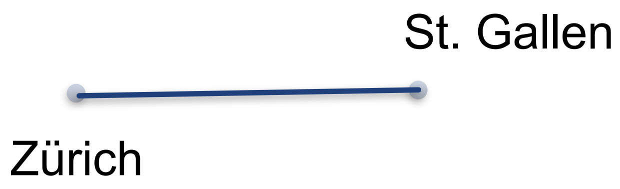
$$O\left(p(n)\cdot 2^{\frac{1}{2}n\log n}\right)$$

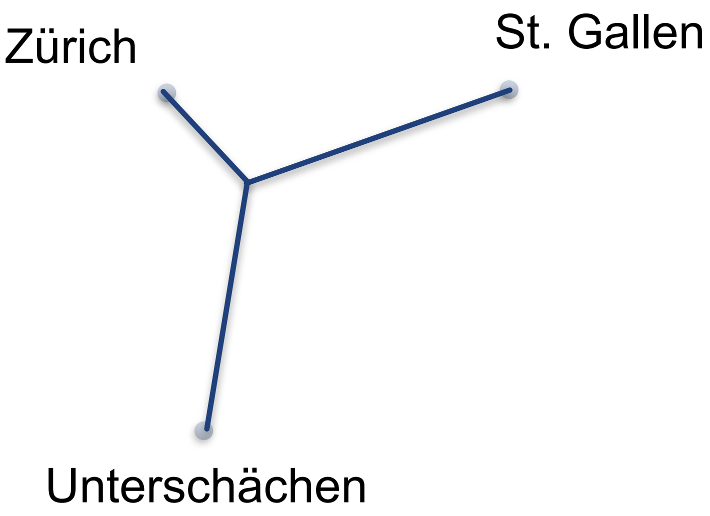
Example 2

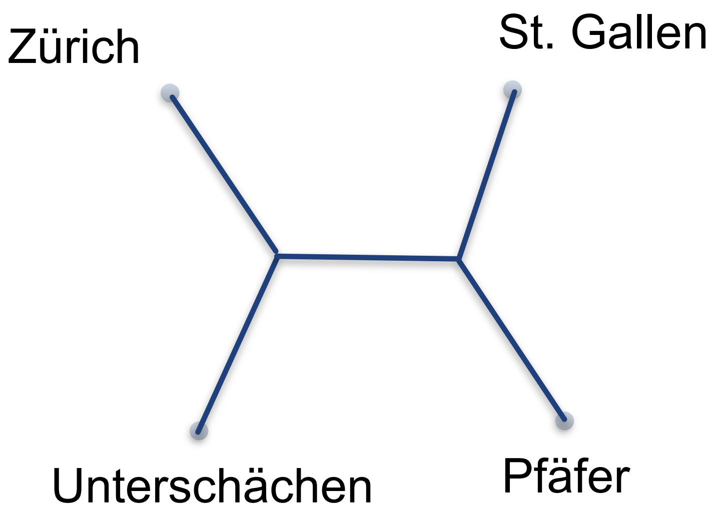
- **How to build the shortest power network?**

Example 2

- **How to build the shortest power network?**





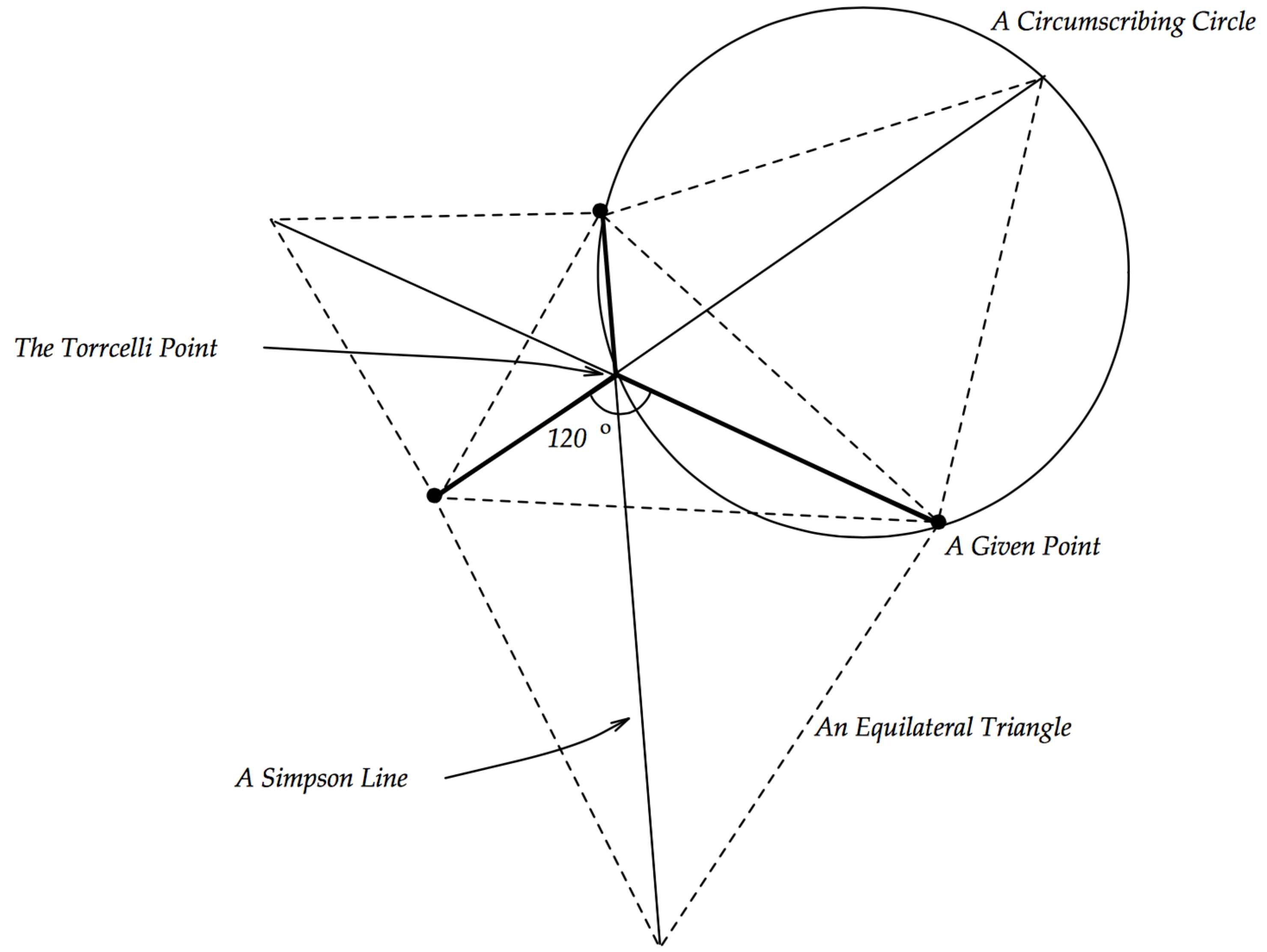


Exampl

- **How to**

Zürich

Z



Example 2

- **Sketchy algorithm**

1. Generate all the inter-station.

   At most n-2 stations are needed. Inter-stations have degree of 3. Others have no more than 3.

2. Build spanning trees together with inter-stations.

3. If shorter substitute the present one till end.

Example 2

- **Quantum implementation**

  "Hardware" design:
  We need $n' = 2n - 2$ of columns and rows of adjacency matrix. Total size $S = n'^2 = 4(n-1)^2$ qubits.

  $\forall \alpha_{ij} \in \{0,1\},\ \alpha_{ij} = \alpha_{ji}$

  $\sum \alpha_{ki} - \alpha_{kk} \in \{0,3\},\ k \geq n+1$

  $\sum \alpha_{ki} - \alpha_{kk} \in \{1,2,3\},\ k < n+1$

column

row

<table>
<tr><th></th><th>1</th><th>2</th><th>…</th><th>n</th><th>n+1</th><th>…</th><th>n’</th></tr>
<tr><td>1</td><td>|-></td><td>|0></td><td>…</td><td>|1></td><td>|1></td><td>…</td><td>|1></td></tr>
<tr><td>2</td><td>|0></td><td>|-></td><td>…</td><td>|0></td><td>|1></td><td>…</td><td>|0></td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td>n</td><td>|1></td><td>|0></td><td>…</td><td>|-></td><td>|0></td><td>…</td><td>|1></td></tr>
<tr><td>n+1</td><td>|1></td><td>|1></td><td>…</td><td>|0></td><td>|-></td><td>…</td><td>|0></td></tr>
<tr><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td><td>…</td></tr>
<tr><td>n’</td><td>|1></td><td>|0></td><td>…</td><td>|1></td><td>|0></td><td>…</td><td>|-></td></tr>
</table>

Example 2

- **Quantum implementation**

"Software" design:
Step 1:
Oracle:

$$O_1|v_1,v_2,...,v_{n'}\rangle = \begin{cases} -|v_1,v_2,...,v_{n'}\rangle & \text{if it is a Steiner tree} \\ |v_1,v_2,...,v_{n'}\rangle & \text{Otherwise} \end{cases}$$

Verifying of Steiner trees:

```
bool ST(*G, V, n){
static bool a=true;
if(n==0){ return a;}
else{if(V>n&&!(childNo==0||childNo==3))
return false;
if(V<=n&&!(childNo==1||2||3))
return false;
if(V.vistit==true) return false:
V.visit==true;
VC=firstchild(V)
while (haschild(V)){
ST(*G, VC, n-1);
VC=nextchild(V):
}
}
}
```

Example 2

- **Applying quantum search algorithm**
  1. Generate equal superposition state and the oracle qubit.

$$|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{v=0}^{N}|v\rangle, \quad |q\rangle = \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

  2. Apply Grover iteration for $R \approx \left\lceil \pi\sqrt{N}/(4\sqrt{M})\right\rceil$ times, where

$$N = 2^{s} = 2^{4(n-1)^2}, \; M < Q \cdot n^{3n} \sim 2^{3n\log n}$$

$$\left[(2|\psi\rangle\langle\psi| - I)O_1\right]^R \frac{1}{\sqrt{N}}\sum_{v=0}^{N}|v\rangle|q\rangle = |S\rangle|q\rangle$$

Example 2

- **Quantum implementation**

Step 2:
We take the Steiner tree ket generated from step1:
Oracle:

$$O_2\left|v_1,v_2,\ldots,v_{n'}\right\rangle=\begin{cases}-\left|v_1,v_2,\ldots,v_{n'}\right\rangle & \text{if it is a minial spanning tree}\\ \left|v_1,v_2,\ldots,v_{n'}\right\rangle & \text{Otherwise}\end{cases}$$

$$\left|v\right\rangle:=\left|v_1,v_2,\ldots,v_n\right\rangle=\bigotimes_{i=1}^{n}\left|v_i\right\rangle=\bigotimes_{i=1}^{n}\bigotimes_{j=1}^{m}\left|\alpha_{ij}\right\rangle,\quad \alpha_{ij}\in\{0,1\}$$

Example 2

- **Applying quantum search algorithm**
  1. Generate equal superposition state and the oracle qubit.

$$|\psi\rangle = |S\rangle, \quad |q\rangle = \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

  2. Apply Grover iteration for $R \approx \lceil \pi\sqrt{N}/4 \rceil$ times, where $N = 2^s = 2^{4(n-1)^2}$

$$\left[(2|\psi\rangle\langle\psi| - I)O_1\right]^R |S\rangle|q\rangle \approx |v_0\rangle|q\rangle$$

  3. Measure! It collapses to v_0.

Example 2

- **Applying quantum search algorithm**

1. Generate equal superposition state and the oracle qubit.

$$|\psi\rangle = |S\rangle, \quad |q\rangle = \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

2. Apply Grover iteration for $R \approx \lceil \pi\sqrt{N}/4 \rceil$ times, where $N = 2^s = 2^{4(n-1)^2}$

$$\left[(2|\psi\rangle\langle\psi| - I)O_1\right]^R |S\rangle|q\rangle \approx |v_0\rangle|q\rangle$$

3. Measure! It collapses to v_0.

Here, we can easily see the complexity is

$$O\left(p(n)\cdot 2^{2(n-1)^2 - 1.5n\log n}\right) \quad \text{NP-hard}$$

Example 2

- **Applying q**

  1. Genera

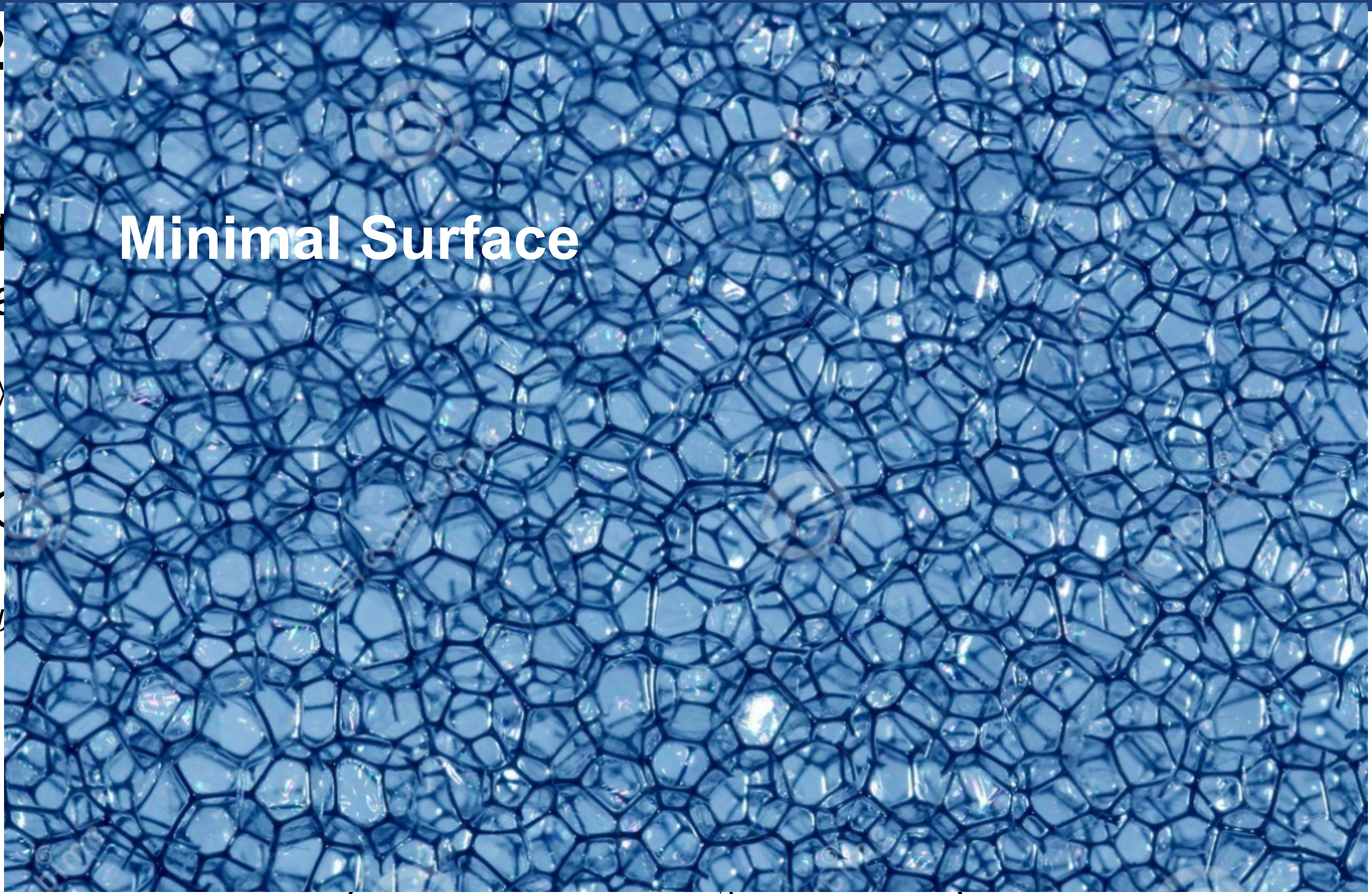
  $|\psi\rangle = |S\rangle$

  2. Apply (

  $[(2|\psi\rangle\langle\psi$

  3. Measu

  Here, we



$$O\left(p(n)\cdot 2^{2(n-1)^2-1.5n\log n}\right)$$ NP-hard

Example 3

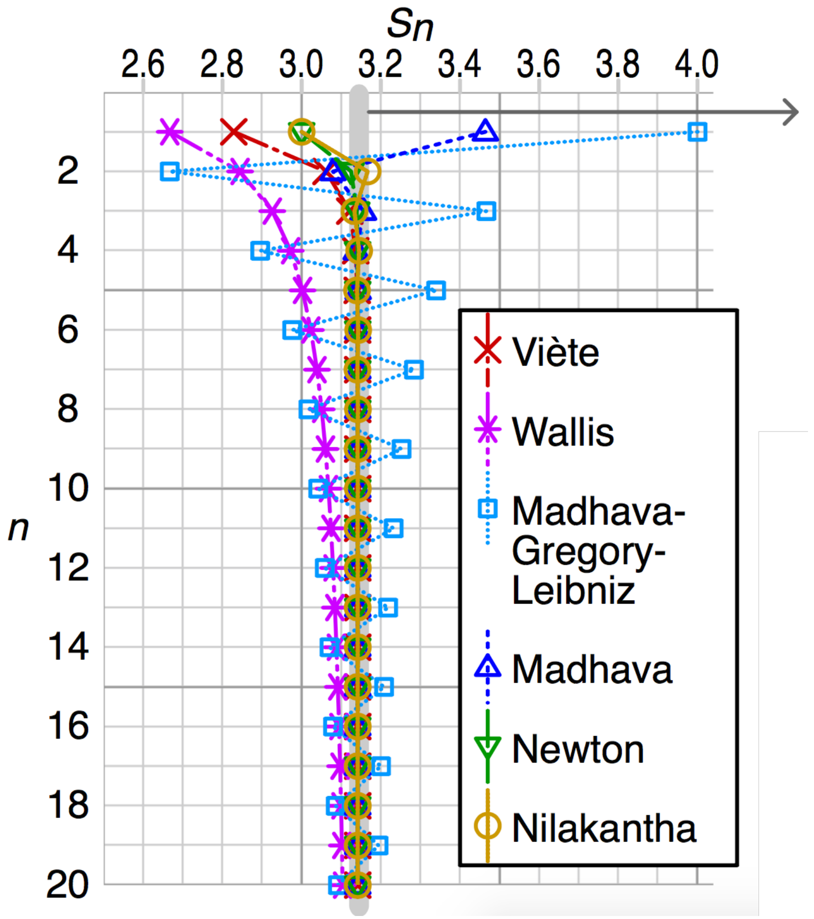
- **Calculate π in billions of digits**

  Taylor

  $$\pi = 4\sum_{k=0}^{\infty}\frac{(-1)^k}{2k+1}$$

  Wallis

  $$\pi = \prod_{k=1}^{\infty}\frac{4k^2}{(2k+1)(2k-1)}$$



- Ramanujan!

  Modular equantion, miracle elliptical integral and ϑ-functions

$$K(k) = \int_0^{\pi/2} \frac{1}{\sqrt{1 - k^2 \sin^2 x}} \mathrm{d}x$$

$$\Rightarrow K(k) = \frac{\pi}{2} \vartheta_3^2(q(k)) \Rightarrow$$

$$\frac{1}{\pi} = \frac{\sqrt{8}}{99^2} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(n!)^{4k} (396)^{4k}}$$

- Ramanujan!

  Modular equantion, miracle elliptical integral and ϑ-functions

$$K(k)=\int_0^{\pi/2}\frac{1}{\sqrt{1-k^2\sin^2 x}}\,\mathrm{d}x$$

$$\Rightarrow K(k)=\frac{\pi}{2}\vartheta_3^2(q(k))\Rightarrow$$

$$a_0=6-4\sqrt{2},\; y_0=\sqrt{2}-1$$

$$y_{n+1}=\frac{1-\sqrt[4]{1-y_n^4}}{1+\sqrt[4]{1-y_n^4}},\; a_{n+1}=(1+y_{n+1})^4 a_n-2^{2n+3}y_{n+1}(1+y+y_{n+1}^2)$$

$$\Rightarrow 0<a_n-1/\pi<16\cdot 4^n e^{-2\cdot 4^n\pi}$$

Example 3

- **Quantum improvement**

  "Hardware" design:
  N - digits of 10-base, wo we need 4 bits for quantum BCD code system. Totally, we have 4N digits.

  "Software" design:
  Oracle:

$$O|x\rangle = \begin{cases} -|x\rangle & \text{if it the series converges} \\ |x\rangle & \text{Otherwise} \end{cases}$$

Index

Block		1	2	3	4
	1	|0>	|0>	…	|0>
	2	|0>	|0>	…	|1>
	…	…	…	…	…
	N	|1>	|1>	…	|1>

Example 3

- **Applying quantum search algorithm**
  1. Generate equal superposition state and the oracle qubit.

  $$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{v=0}^{N} |x\rangle, \quad |q\rangle = \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

  2. Apply Grover iteration for $R \approx \lceil \pi\sqrt{N}/4 \rceil$ times.

  $$\left[(2|\psi\rangle\langle\psi| - I)O\right]^R \frac{1}{\sqrt{N}} \sum_{v=0}^{N} |x\rangle|q\rangle \approx |x\rangle|q\rangle$$

  3. Measure! It collapses to v_0.

Quantum complexity: $O\left(p(N)\cdot\sqrt{N}\right)$

Classical complexity: $O\left(p(N)\cdot N\right)$

Oracle complexity:

$$p(N) = \log(N)\log\left(\log(N)\right)$$

Optimality

- **Faster Search algorithm?**

Under the framework we are discussing the answer ist “NO”!

Sketchy proof by induction and Cauchy-Schwarz-inequality:

$$\left|\psi_x^k\right\rangle \equiv U_k O_x U_{k-1} O_x \dots U_1 O_x \left|\psi\right\rangle$$

$$\left|\psi_x\right\rangle \equiv U_k U_{k-1} \dots U_1 \left|\psi\right\rangle$$

$$D_k \equiv \sum \left\| \left|\psi_x^k\right\rangle - \left|\psi_x\right\rangle \right\|^2 \leq 4k^2$$

$$D_{k+1} \equiv \sum \left\| O_x \left|\psi_x^k\right\rangle - \left|\psi_x\right\rangle \right\|^2$$

$$= \sum \left\| O_x \left(\left|\psi_x^k\right\rangle - \left|\psi_x\right\rangle \right) + (O_x - I)\left|\psi_x\right\rangle \right\|^2$$

$$D_{k+1} = D_k + 4\sqrt{D_k} + 4 < 4(k+1)^2$$

$$D_k \geq \sum \left\| \left|\psi_k^x\right\rangle - \left|x\right\rangle \right\|^2 + \sum \left\| \left|\psi_k\right\rangle - \left|x\right\rangle \right\|^2 - 2\sum \left\| \left|\psi_k\right\rangle - \left|x\right\rangle \right\| \cdot \left\| \left|\psi_k^x\right\rangle - \left|x\right\rangle \right\|$$

$$D_k \geq \left(\sqrt{\sum \left\| \left|\psi_k\right\rangle - \left|x\right\rangle \right\|^2} - \sqrt{\sum \left\| \left|\psi_k^x\right\rangle - \left|x\right\rangle \right\|^2} \right)^2 \sim QN$$

$$k = \Omega\left(\sqrt{N}\right)$$

$$\Rightarrow k = \Theta\left(\sqrt{N}\right)$$

Thanks for listening!!

References:

[1] R. L. Graham, D. A. Thomas, and M. Zachariasen: Euclidean Steiner Tree Problem

[2] P. Jaillet: MIT Lecture notes on Introduction to Algorithms

[3] J. M. Borwein and P. R. Borwein: Ramanujan, Modular Equations, and Approximations to pi

[4]vA. Nielsen and L. Chuang: Quantum Computation and Quantum Information

[5] Wikipedia: Steiner tree, Elliptical integral.