The Grover’s Algorithm

Camilo Zapata
Xiao Yang
The problem

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

\[ \text{Entries} = N = 2^n \]

<table>
<thead>
<tr>
<th>Person</th>
<th>Height</th>
<th>Age</th>
<th>Weight</th>
<th>IQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>173</td>
<td>25</td>
<td>200</td>
<td>95</td>
</tr>
<tr>
<td>Peter</td>
<td>175</td>
<td>26</td>
<td>185</td>
<td>75</td>
</tr>
<tr>
<td>Greg</td>
<td>185</td>
<td>32</td>
<td>191</td>
<td>95</td>
</tr>
<tr>
<td>James</td>
<td>165</td>
<td>28</td>
<td>180</td>
<td>150</td>
</tr>
<tr>
<td>Matthew</td>
<td>162</td>
<td>15</td>
<td>140</td>
<td>136</td>
</tr>
<tr>
<td>Peter</td>
<td>145</td>
<td>12</td>
<td>130</td>
<td>100</td>
</tr>
</tbody>
</table>

Find element on a database.

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

$|\psi\rangle$ input $\xrightarrow{G^k} |x_0\rangle$ output
Breaking it down

In the case where \( n = 2 \), \( N = 2^n = 4 \)

\[
|\psi\rangle = \frac{1}{2} (|00\rangle + |01\rangle + |10\rangle + |11\rangle)
\]
Breaking it down

The Grover's Algorithm

\[ \begin{array}{c}
\text{n qubits } |0\rangle \\
\text{oracle workspace}
\end{array} \]

\[ H^\otimes n \]

\[ G \]

\[ O(\sqrt{N}) \]

\[ \text{measure} \]

\[ \ldots \]
Grover’s operator: step 1

- The oracle:

If the solution for search is $x_0$, then:

$$f(x) = \begin{cases} 1 & \text{if } x = x_0 \\ 0 & \text{otherwise} \end{cases},$$

$$|x\rangle \xrightarrow{O} (-1)^{f(x)}|x\rangle$$

Example $N=4$

Solution $x_0 = 2$

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

$$O \xrightarrow{\text{Phase:}} \begin{cases} |0\rangle \rightarrow |0\rangle \\ |x\rangle \rightarrow -|x\rangle \\ \text{for } x > 0 \end{cases}$$

$$O \xrightarrow{\text{Phase:}} \frac{1}{2} (|00\rangle + |01\rangle - |10\rangle + |11\rangle)$$
Grover’s operator: step 2

Example N=4

Solution $x_0 = 2$

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

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

$$H^\otimes 2 \rightarrow \frac{1}{2} (|00\rangle - |01\rangle + |10\rangle + |11\rangle)$$
Grover’s operator: step 3

Example $N=4$

Solution $x_0 = 2$

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

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

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

$$\text{phase} \rightarrow \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle)$$
N=4 is a special case. We get to the desired state with just one iteration of the Grover’s algorithm.
The 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

\[ G = (2 \left| \psi \right\rangle \langle \psi \right| - I) O \]

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

A reflection around the \( \left| \alpha \right\rangle \) 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:

\[ (2 |\psi\rangle \langle \psi| - I) |\phi\rangle = 2 \langle \psi |\phi\rangle |\psi\rangle - |\phi\rangle \]
Geometric interpretation

\[ G = \langle 2 \psi | \psi \rangle - I \] \[ O \]

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

\[ \langle 2 \psi | \psi \rangle - I \rangle \phi = 2 \langle \psi | \phi \rangle \psi - \phi \]
Geometric interpretation

\[ G = (2 \left| \psi \right\rangle \left\langle \psi \right| - I) O \]

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

\[ (2 \left| \psi \right\rangle \left\langle \psi \right| - I) \left| \phi \right\rangle = 2 \left( \left| \psi \right\rangle \left\langle \phi \right| \left. \phi \right\rangle \right) \left| \psi \right\rangle - \left| \phi \right\rangle \]
Geometric interpretation

\[ G = (2 \langle \psi | \psi \rangle - I) O \]

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

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

A reflection around the \(|\psi\rangle\) state.
Geometric interpretation

Therefore:

\[ G = (2\ket{\psi}\bra{\psi} - I)O \]

is a rotation.

Then:

\[ G\ket{\psi} = \cos \frac{3\theta}{2} \ket{\alpha} + \sin \frac{3\theta}{2} \ket{\beta}, \quad \cos \left( \frac{\theta}{2} \right) = \sqrt{\frac{N - 1}{N}} \]

\[ G^k\ket{\psi} = \cos \left( \frac{2k + 1}{2} \theta \right) \ket{\alpha} + \sin \left( \frac{2k + 1}{2} \theta \right) \ket{\beta} \]
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:

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

$$R \leq \left\lfloor \frac{\pi}{4} \sqrt{\frac{N}{M}} \right\rfloor$$

So,

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

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

$$R \leq \left[ \frac{\pi}{4} \sqrt{\frac{2N}{M}} \right]$$

So,

$$R = O\left(\sqrt{\frac{N}{M}}\right)$$
Performance of quantum algorithm

Complexity

- space complexity
- **time complexity**

\[
\begin{align*}
f(x) &= O(g(x)) \quad f(x) \leq k|g(x)| \\
f(x) &= \Omega(g(x)) \quad f(x) \geq k|g(x)| \\
f(x) &= \Theta(g(x)) \quad n|g(x)| \leq f(x) \leq m|g(x)| \\
\exists \quad n, m, k \in \mathbb{R}^+ 
\end{align*}
\]

Grover’s algorithm

\[
R \leq \left[ \frac{\pi}{4 \sqrt{M}} \right] = O(\sqrt{N})
\]

- 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**.

\[ P \subseteq NP \]
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.

\[ P \subseteq NP \]
P=NP? conjecture

- **NPC** a subset of **NP**
  
  Every problem in **NP** is **reducible in polynomial time** 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 $O(p(n) \cdot 2^{n \log n})$
- generate each possible ordering $(v_1, v_2, \ldots, v_n)$.
- check each ordering for whether it is Hamiltonian cycle.
Example 1

- **Hamiltonian Cycle**

Classically, it has complexity of \( O(p(n) \cdot 2^{n \log n}) \)
- generate each possible ordering \((v_1, v_2, ..., v_n)\).
- check each ordering for whether it is Hamiltonian cycle.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>X</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>123</th>
<th>231</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>132</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>312</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>213</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>321</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
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:
  
  $|v\rangle = |v_1, v_2, \ldots, v_n\rangle = \bigotimes_{i=1}^{n} |v_i\rangle = \bigotimes_{i=1}^{n} \bigotimes_{j=1}^{m} |\alpha_{ij}\rangle, \quad \alpha_{ij} \in \{0, 1\}$
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 \( R \approx \left\lceil \frac{\pi \sqrt{N}}{4} \right\rceil \) times, where \( N = 2^S = 2^{mn} = 2^{n \log[n]} \).

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

  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 \( R \approx \left\lfloor \frac{\pi \sqrt{N}}{4} \right\rfloor \) times, where \( N = 2^S = 2^{mn} = 2^{n \log n} \).
     \[
     \left[ (2|\psi\rangle\langle\psi| - I)O \right]^R \frac{1}{\sqrt{N}} \sum_{v=0}^{N} |v\rangle|q\rangle \approx |v_0\rangle|q\rangle
     \]

  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?

Zürich → St. Gallen

Zürich → St. Gallen

Unterschächen

Zürich → Unterschächen → Pfäfer

Zürich → St. Gallen

Example

- How to build the shortest power network?

Zürich
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 $n^1 = 2n - 2$ of columns and rows of adjacency matrix. Total size $S = n^1^2 = 4(n - 1)^2$ qubits.

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

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

$\sum \alpha_{ki} - \alpha_{kk} \in \{1, 2, 3\}, k < n + 1$
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||2|3))
  return false;
  if(V.visit==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 \( R \approx \left\lfloor \frac{\pi \sqrt{N}}{4\sqrt{M}} \right\rfloor \) times, where

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

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

- Quantum implementation

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

\[
O_2 |v_1, v_2, ..., v_n\rangle = \begin{cases} 
- |v_1, v_2, ..., v_n\rangle & \text{if it is a minimal spanning tree} \\
|v_1, v_2, ..., v_n\rangle & \text{Otherwise}
\end{cases}
\]

\[
|\psi\rangle := |v_1, v_2, ..., v_n\rangle = \bigotimes_{i=1}^{n} |v_i\rangle = \bigotimes_{i=1}^{n} \bigotimes_{j=1}^{m} |\alpha_{ij}\rangle, \quad \alpha_{ij} \in \{0, 1\}
\]
Example 2

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

\[ |\psi\rangle = |S\rangle, \quad |q\rangle = \left[ \begin{array}{c} 1 \\ -1 \\ \sqrt{2} \end{array} \right] \]

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

\[ \left[ 2|\psi\rangle\langle\psi| - I \right]_O \left| S \rangle \langle q \right| \approx \left| v_0 \rangle \langle q \right| \]

3. Measure! It collapses to \( v_0 \).
Applying quantum search algorithm

1. Generate equal superposition state and the oracle qubit.
\[ |\psi\rangle = |S\rangle, \quad |q\rangle = \left| \begin{array}{c} 0 \\ -1 \\ \sqrt{2} \end{array} \right| \]

2. Apply Grover iteration for \( R \approx \left\lceil \pi \sqrt{N/4} \right\rceil \) times, where \( N = 2^s = 2^{4(n-1)^2} \)
\[ \left( 2|\psi \rangle \langle \psi | - I \right)^R |S\rangle |q\rangle \approx |v_0\rangle |q\rangle \]

3. Measure! It collapses to \( v_0 \).

Here, we can easily see the complexity is
\[ O\left( p(n) \cdot 2^{2(n-1)^2 - 1.5n \log n} \right) \] NP-hard
Example 2

- Applying q-

1. Generate equal superposition state and the oracle qubit.

\[ |\psi\rangle = |S\rangle \]

2. Apply Grover iteration for

\[ N = \frac{\pi}{2} \sqrt{\frac{N}{2}} \]

3. Measure! It collapses to

\[ |v_0\rangle \]

\[ \approx \begin{bmatrix} 0 \\ -1/2 \end{bmatrix} \]

Here, we can easily see the complexity is

\[ O(p(n) \cdot 2^{2(n-1)^2-1.5n\log n}) \]

NP-hard
Example 3

- Calculate $\pi$ 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 equation, miracle elliptical integral and \(\vartheta\)-functions

\[
K(k) = \int_{0}^{\pi/2} \frac{1}{\sqrt{1 - k^2 \sin^2 x}} \, dx
\]

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

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

Modular equation, miracle elliptical integral and \( \vartheta \)-functions

\[
K(k) = \int_{0}^{\pi/2} \frac{1}{\sqrt{1 - k^2 \sin^2 x}} \, dx
\]

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

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

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

\[
\Rightarrow 0 < a_n - 1/\pi < 16 \cdot 4^n e^{-2^{4n} \pi}
\]
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:
  \[ O|x\rangle = \begin{cases} 
  -|x\rangle & \text{if the series converges} \\
  |x\rangle & \text{Otherwise} 
  \end{cases} \]

  \[
  \begin{array}{cccc}
  \text{Index} & 1 & 2 & 3 & 4 \\
  1 & |0\rangle & |0\rangle & \ldots & |0\rangle \\
  2 & |0\rangle & |0\rangle & \ldots & |1\rangle \\
  \ldots & \ldots & \ldots & \ldots & \ldots \\
  N & |1\rangle & |1\rangle & \ldots & |1\rangle 
  \end{array}
  \]
Example 3

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

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

2. Apply Grover iteration for \( R \approx \left\lfloor \frac{\pi \sqrt{N}}{4} \right\rfloor \) times.

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

3. Measure! It collapses to \( v_0 \).

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

**Classical complexity:** \( O(p(N) \cdot N) \)

**Oracle complexity:**
\[
p(N) = \log(N) \log(\log(N))
\]
Optimality

- Faster Search algorithm?

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

Sketchy proof by induction and Cauchy-Schwarz-inequality:

\[
\begin{align*}
|\psi_x^k\rangle &\equiv U_k O_x U_{k-1} O_x \cdots U_1 O_x |\psi\rangle & D_{k+1} &= D_k + 4\sqrt{D_k} + 4 < 4(k+1)^2 \\
|\psi_x\rangle &\equiv U_k U_{k-1} \cdots U_1 |\psi\rangle \\
D_k &\equiv \sum \| |\psi_{x}^k\rangle - |\psi_{x}\rangle \|^2 \leq 4k^2 \\
D_{k+1} &\equiv \sum \left\| O_x (|\psi_{x}^k\rangle - |\psi_{x}\rangle) + (O_x - I)|\psi_{x}\rangle \right\|^2 \\
&= \sum \left\| O_x (|\psi_{x}^k\rangle - |\psi_{x}\rangle) \right\|^2 - \Omega \left( \sqrt{N} \right) \\
&\Rightarrow k = \Theta \left( \sqrt{N} \right)
\end{align*}
\]
Thanks for listening!!

References:


