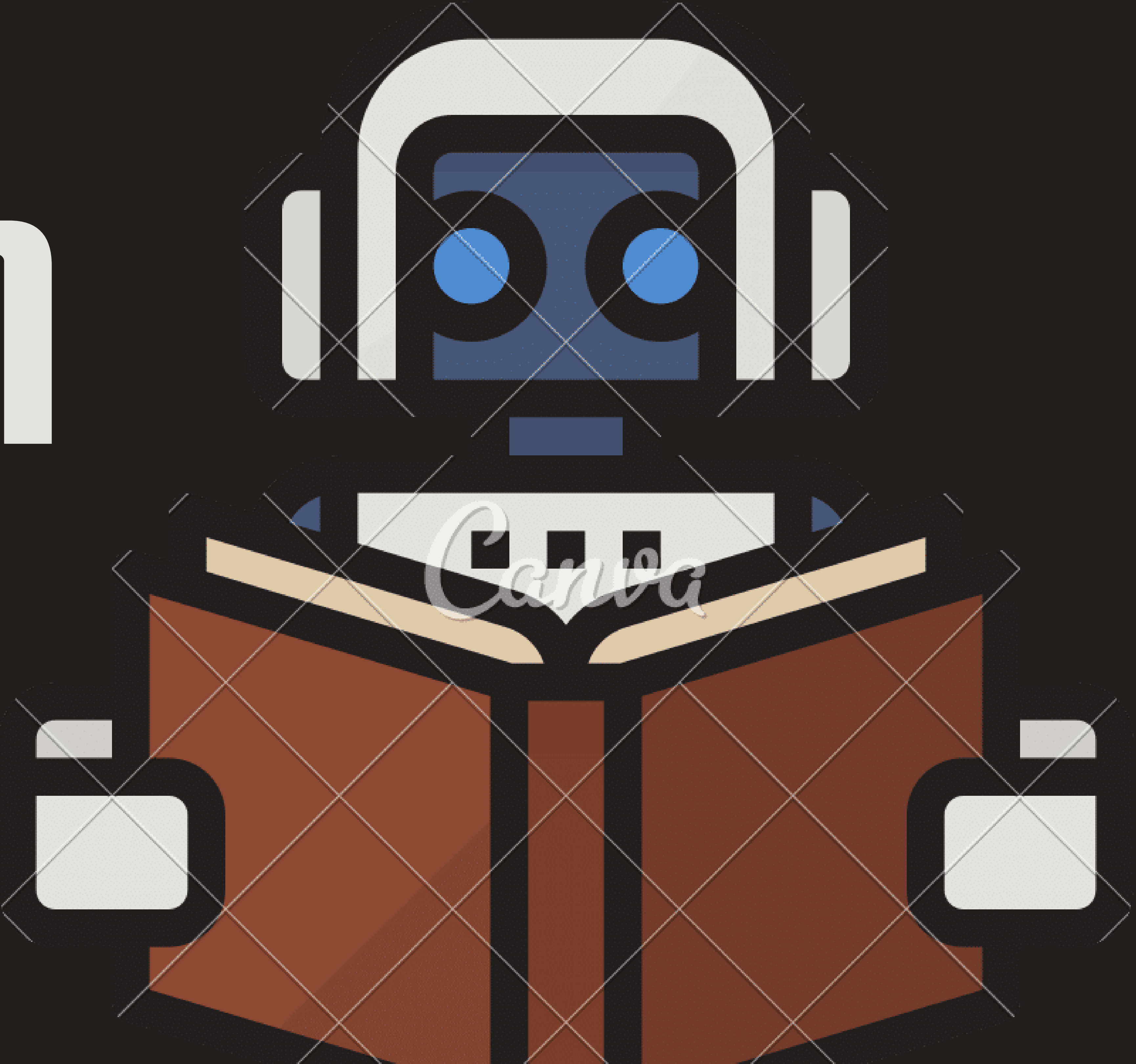
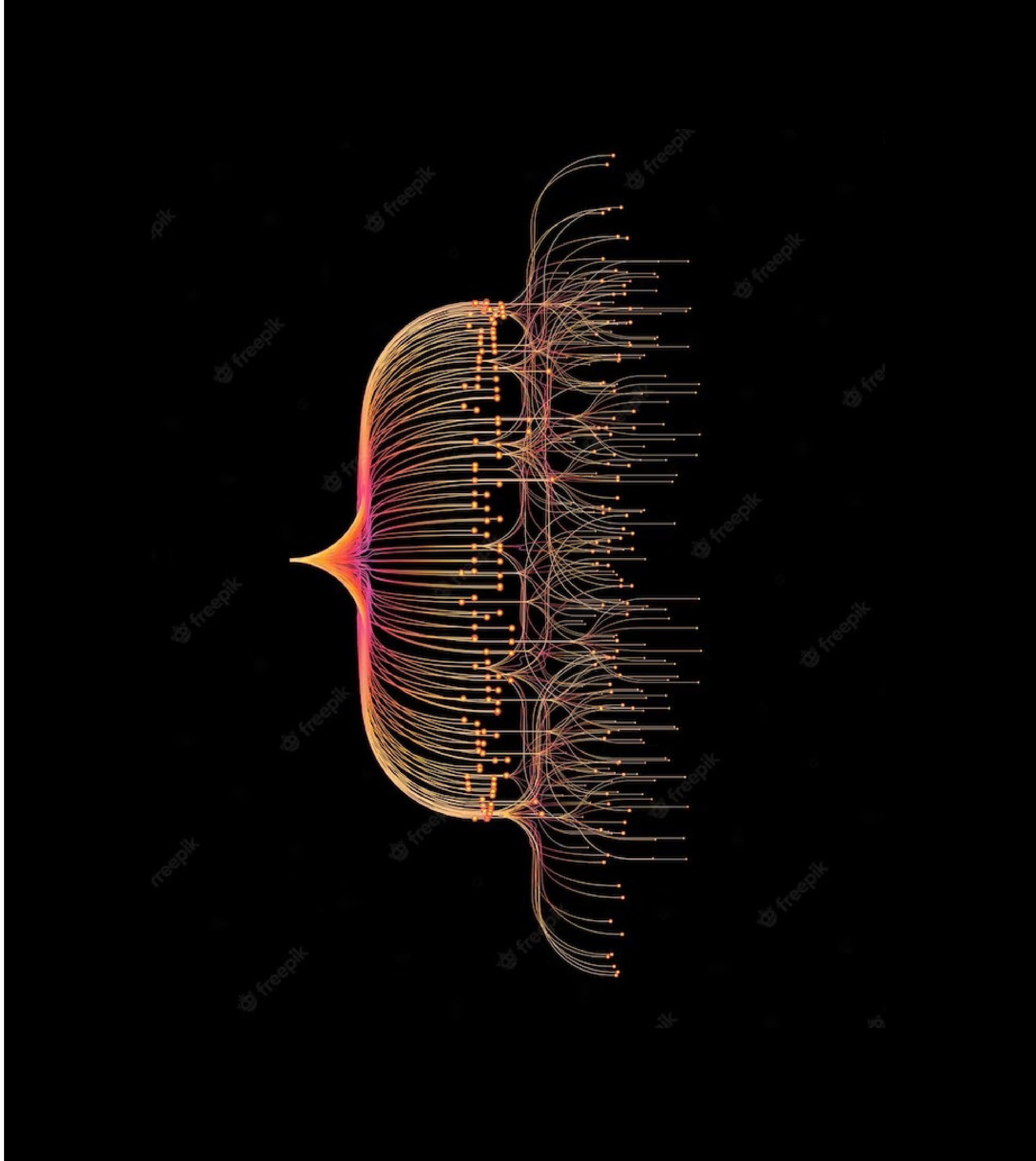
GROVER'S ALGORITHM

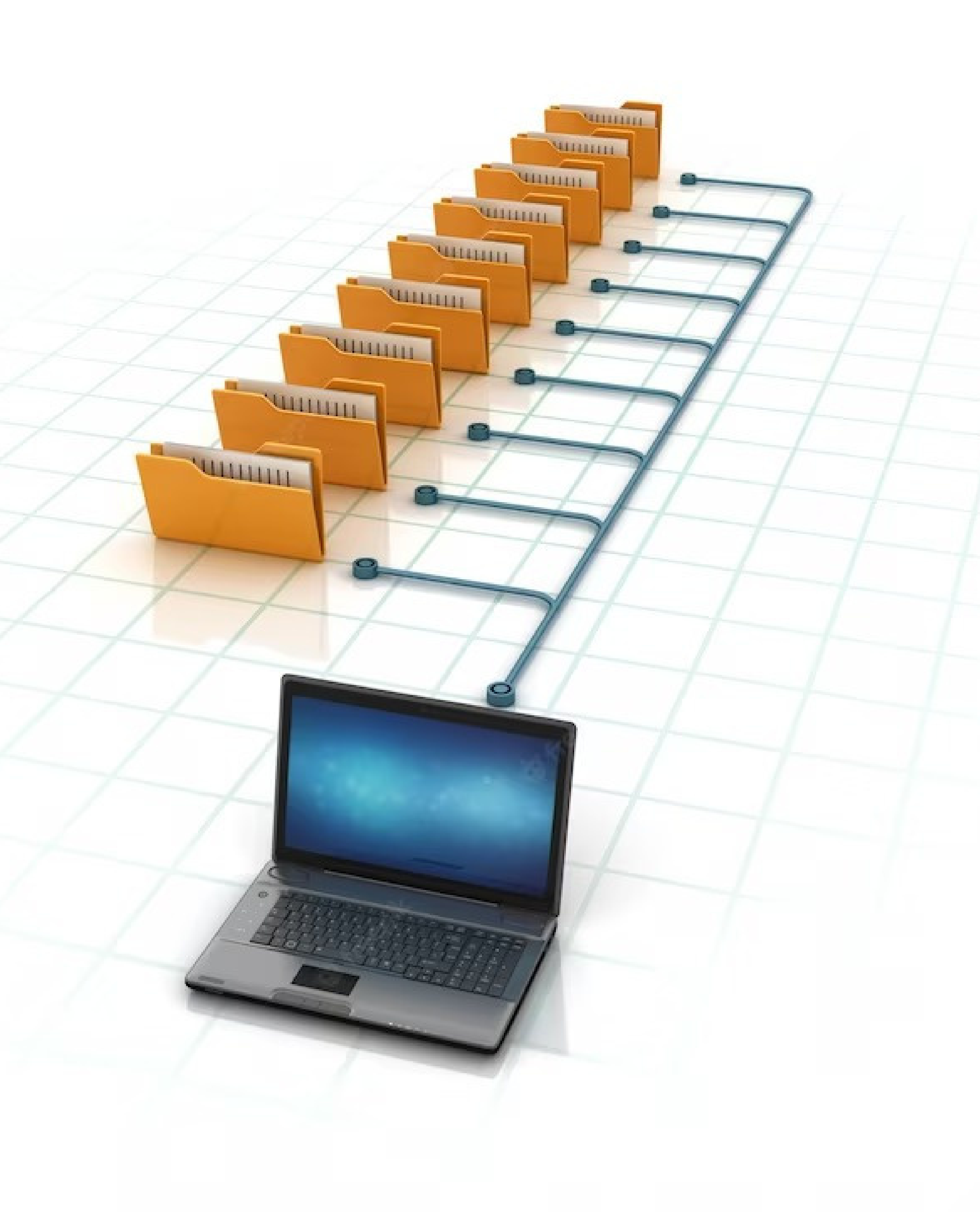
By: Rachit Bhalla
21BAI1869



Introduction

Grover's algorithm is a revolutionary approach to **search**. This quantum algorithm offers exponential speedup over classical search algorithms, opening up new possibilities in data analysis and optimization. In this presentation, we will explore the inner workings of Grover's algorithm and its potential applications.





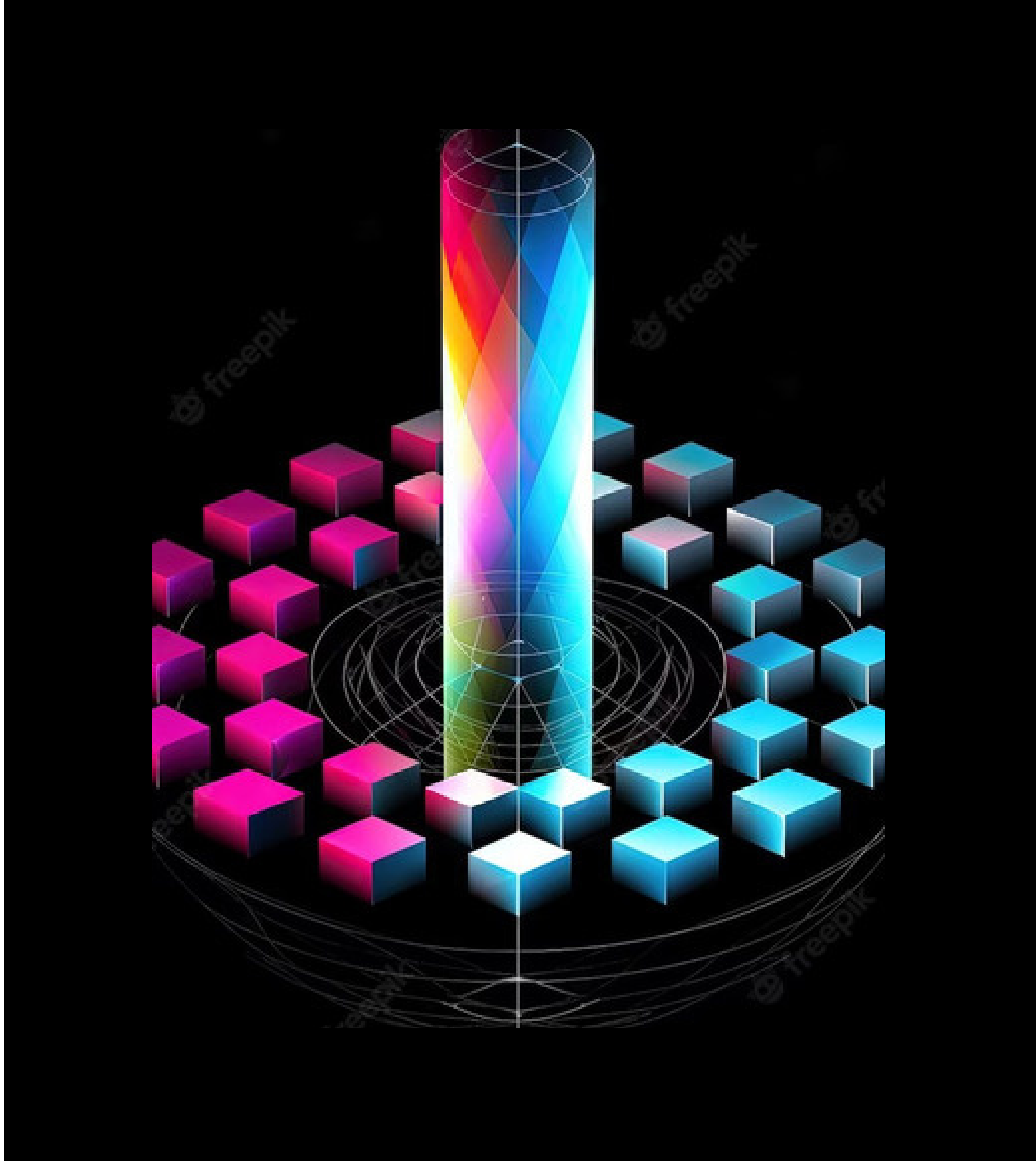
Classical vs. Quantum Search

Classical search algorithms have a time complexity of **O(N)**, where N is the number of elements in the search space. In contrast, Grover's algorithm has a time complexity of **O(√N)**, significantly reducing the number of required iterations. This quantum advantage stems from **quantum superposition** and **interference**, allowing for parallel computation. Grover's algorithm is a game-changer in search problems.

Where N=2^n

Grover's Algorithm Explained

Grover's algorithm leverages **quantum amplitude amplification** to enhance the probability of finding the desired solution. It involves a series of **quantum operations** such as **Oracle**, **Hadamard**, and **Phase Inversion** gates. By iteratively applying these operations, the algorithm amplifies the amplitude of the target state, leading to a higher probability of measurement. This process provides a significant speedup in search tasks.



Algorithm:

1. Initialization: Prepare the quantum state that represents a uniform superposition of all possible solutions. This can be achieved by applying a Hadamard gate to each qubit.
2. Oracle: Apply an oracle operator that marks the target solution. The oracle operator flips the phase of the target solution while leaving the other solutions unchanged.
3. Diffusion: Apply a diffusion operator that amplifies the amplitude of the target solution and suppresses the amplitudes of the other solutions. The diffusion operator reflects the amplitudes about the mean, essentially concentrating the amplitudes around the target solution.

Steps 2 and 3 are repeated for a certain number of iterations, which is approximately equal to sqrt(N) to maximize the probability of finding the target solution.

Illustration:

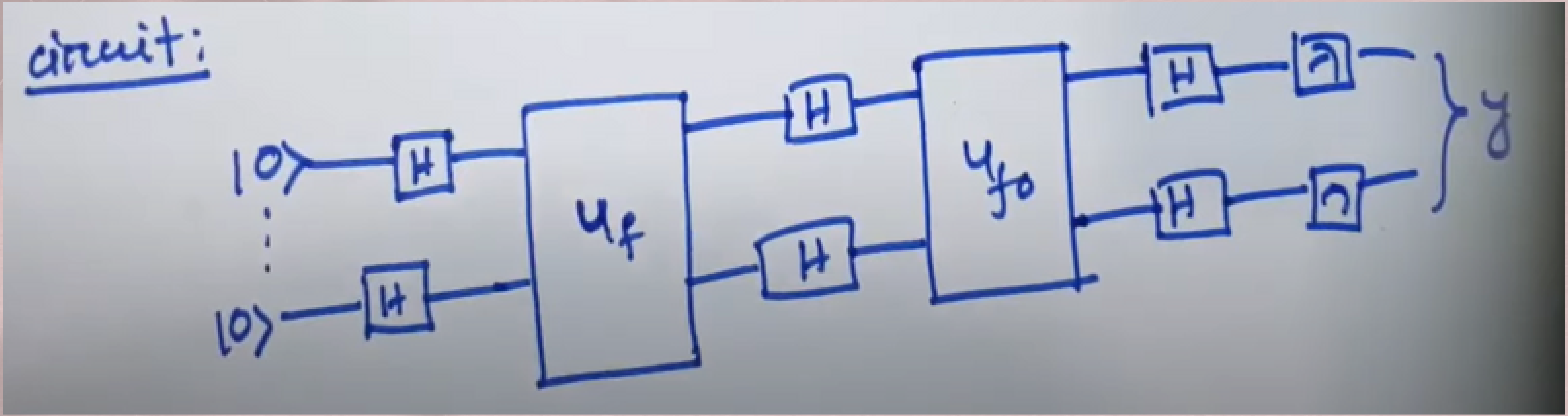
Grover's algorithm for a search problem with 8 elements, where one element is marked as the target.

1. Initialization:
    - We start with 3 qubits, represented by the state $|000\rangle$.
    - Apply a Hadamard gate (H) to each qubit to create a uniform superposition: $H|000\rangle = 1/\text{sqrt}(8) * (|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle)$.
2. Oracle:
    - Apply an oracle operator that marks the target solution. Let's say the target is $|101\rangle$.
    - Apply a phase inversion (Z) gate to the target state: $Z|101\rangle = -|101\rangle$.
    - The oracle operator flips the sign of the target state while leaving the other states unchanged.

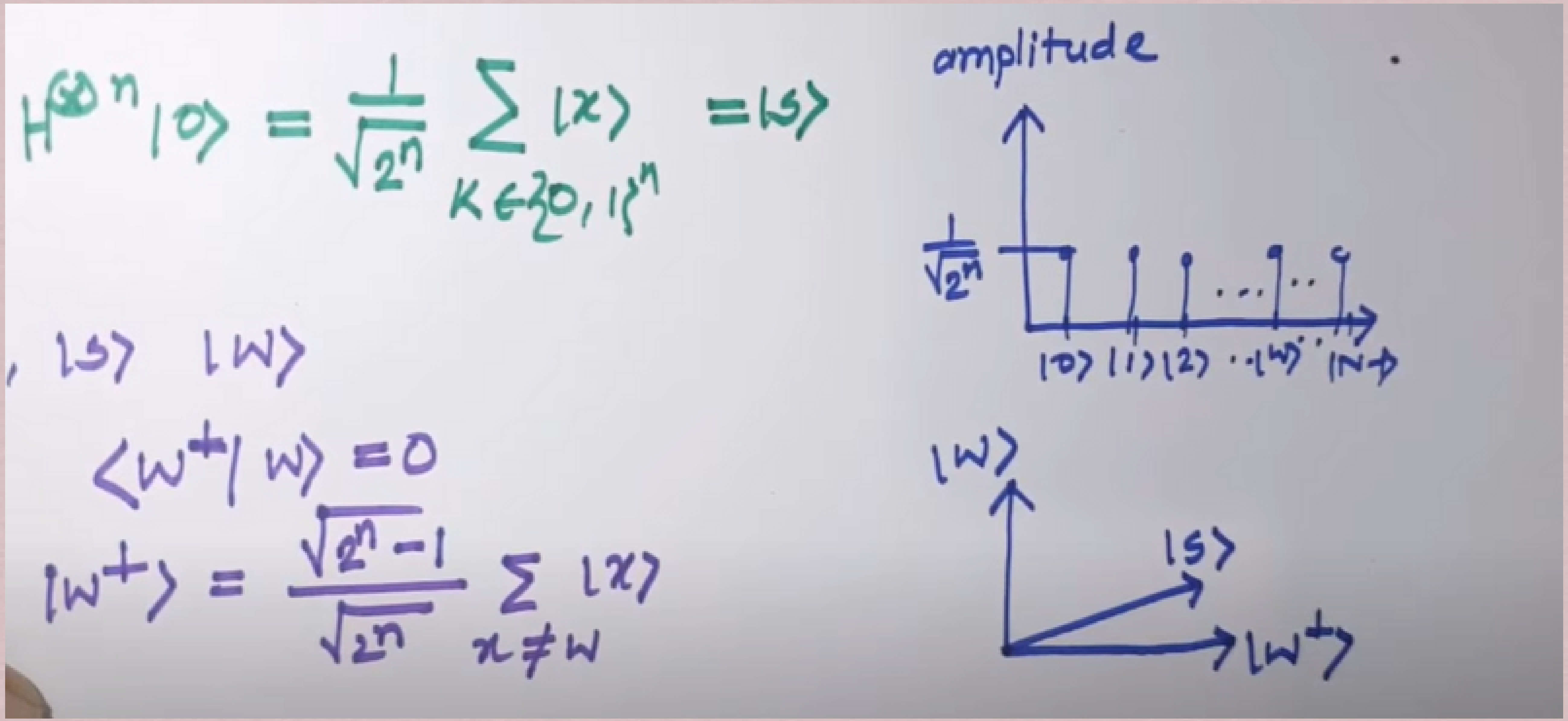
- Diffusion:
  - Apply the diffusion operator, which consists of a combination of Hadamard (H) and phase inversion (Z) gates.
  - First, apply a Hadamard gate (H) to each qubit: H|000⟩ = 1/sqrt(8) * (|000⟩ + |001⟩ + |010⟩ + |011⟩ + |100⟩ + |101⟩ + |110⟩ + |111⟩).
  - Apply a phase inversion (Z) gate to each qubit: Z|000⟩ = |000⟩, Z|001⟩ = |001⟩, Z|010⟩ = |010⟩, Z|011⟩ = |011⟩, Z|100⟩ = -|100⟩, Z|101⟩ = -|101⟩, Z|110⟩ = |110⟩, Z|111⟩ = |111⟩.
  - Apply the Hadamard gate (H) again to each qubit: H|000⟩ = (1/sqrt(8) * (|000⟩ + |001⟩ + |010⟩ + |011⟩ + |100⟩ + |101⟩ + |110⟩ + |111⟩).

- Repeat steps 2 and 3 multiple times:
  - By repeating steps 2 and 3, the amplitudes of the marked state(s) get amplified while the other states are suppressed.
  - The number of iterations needed is approximately sqrt(N), where N is the number of elements (in this case, 8).
- Measurement:
  - After the desired number of iterations, perform a measurement to obtain the final result.
  - The measurement collapses the quantum state into a classical state, giving us one of the possible outcomes.
  - The probability of measuring the target state is increased compared to the other states, improving the chances of finding the solution.

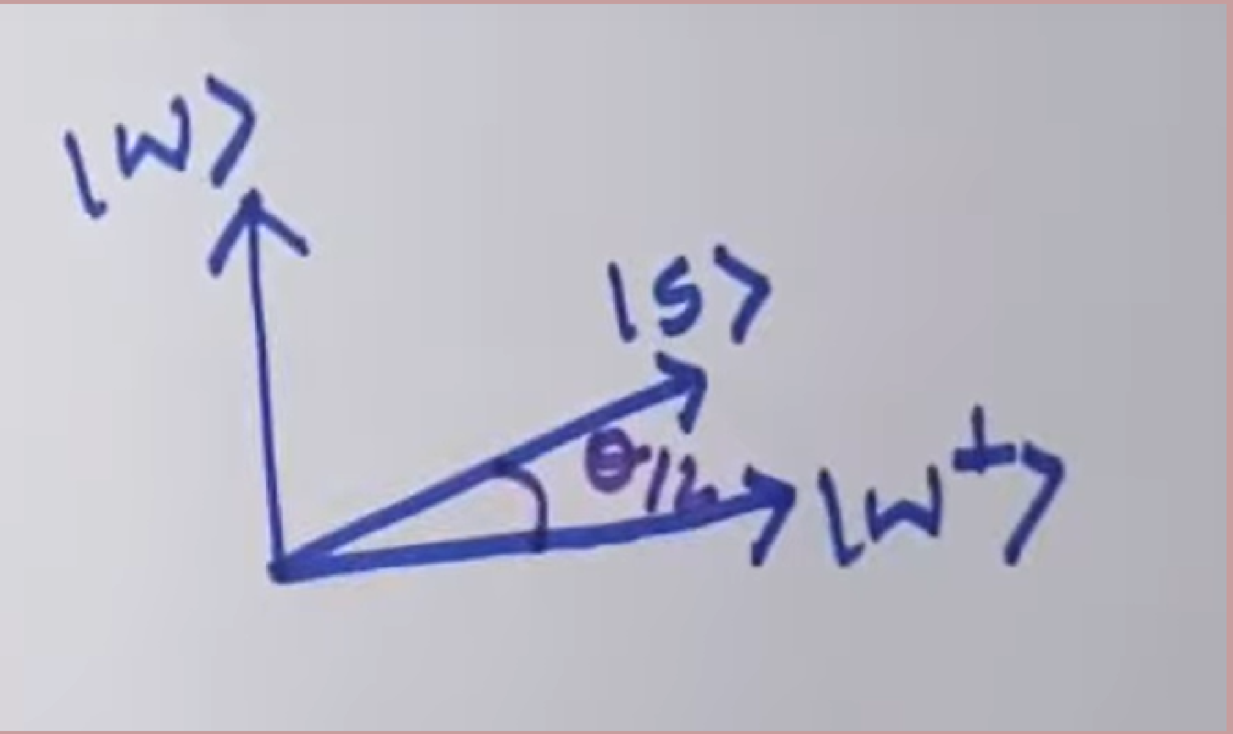
Visualization:



Where Both the oracle functions do inversion and Uf0 does amplification as well which helps in increasing the probability of finding our result faster.



Here we pass our Qubits through Hadamard Gates and hence get the |s> values for each qubit. and then plot it against |w>(orthogonal).



$$|s\rangle = \frac{\sqrt{2^n - 1}}{\sqrt{2^n}}\,|w^{\perp}\rangle + \frac{1}{\sqrt{2^n}}\,|w\rangle$$

$$= \cos\frac{\theta}{2}\,|w^{\perp}\rangle + \sin\frac{\theta}{2}\,|w\rangle$$

$$\theta = 2 \arcsin\left(\frac{1}{\sqrt{2^n}}\right)$$

S is the uniform superposition state for all the Qubits.

(Diagram in previous slide).

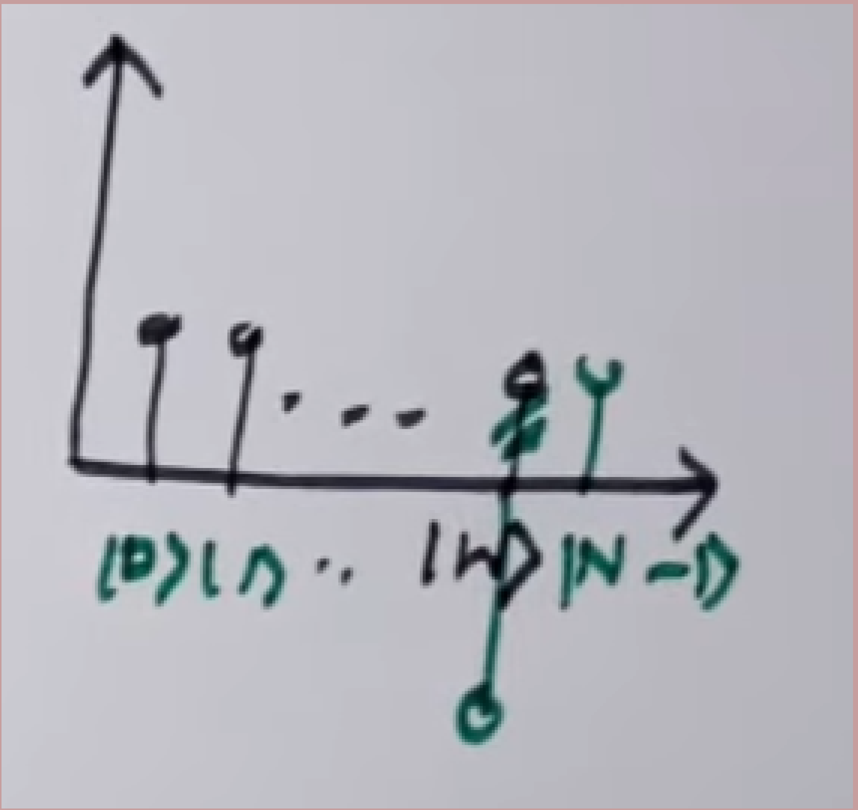
step-02 : Phase inversion

$|w\rangle$, $|s\rangle$ — same

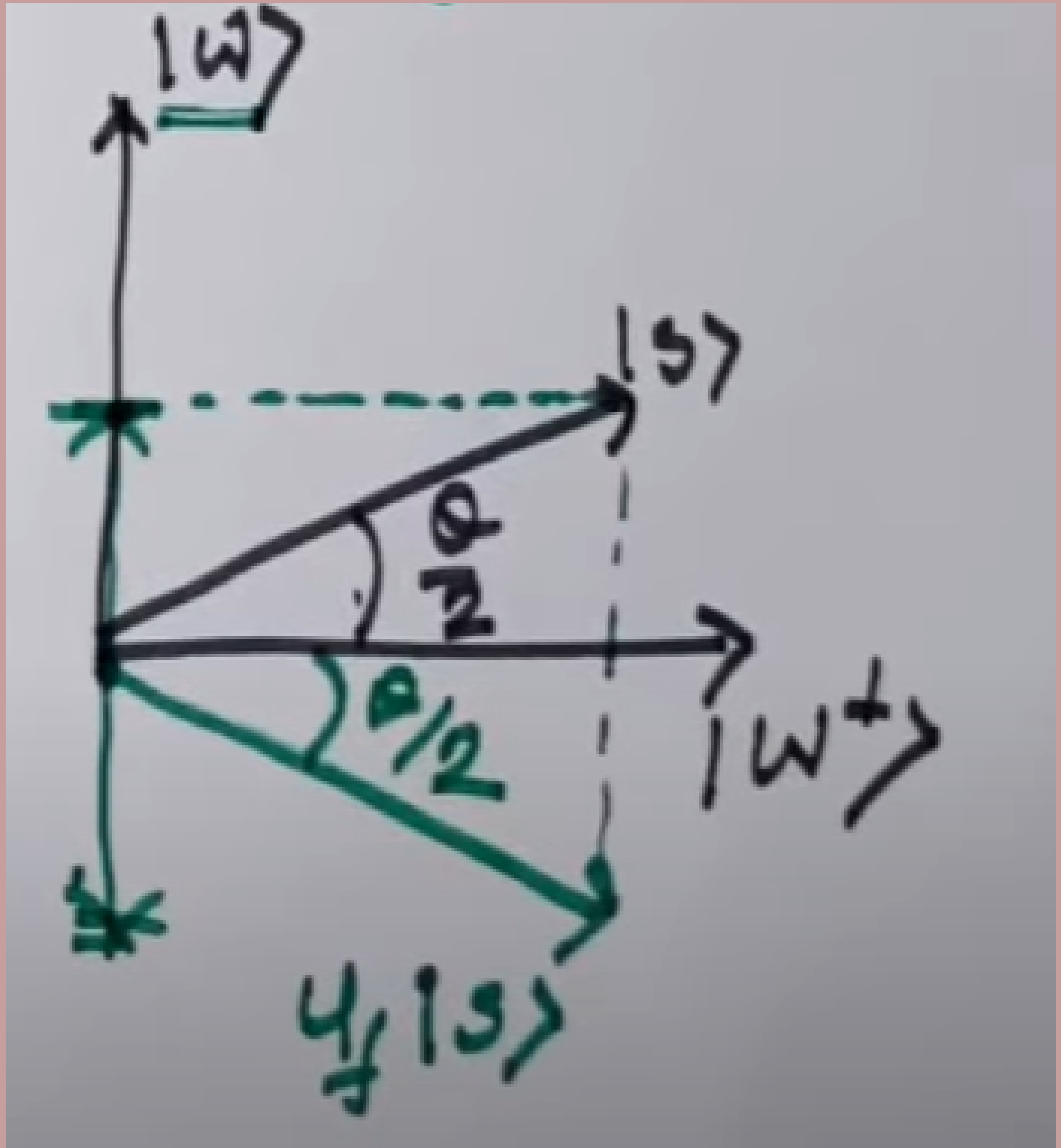
$s \neq w$

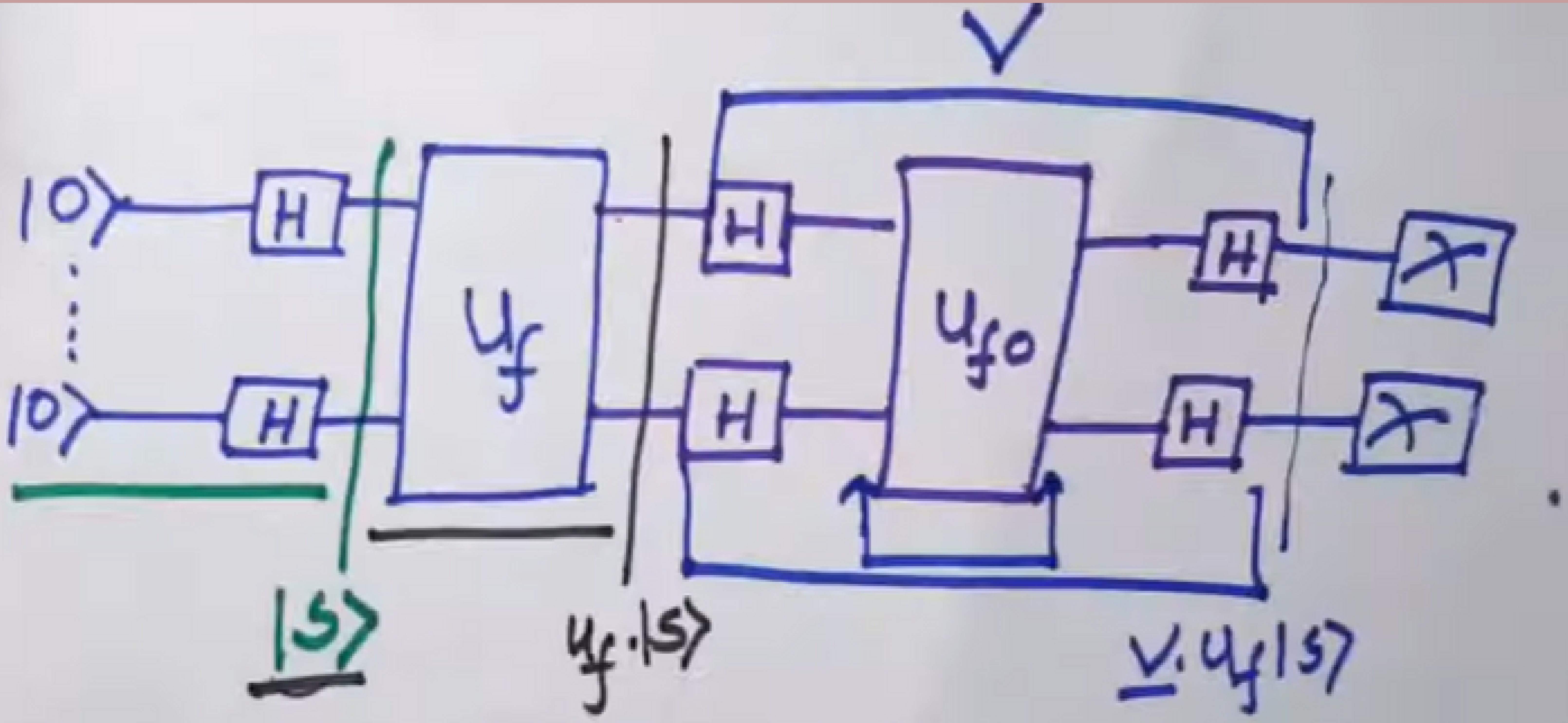
$$U_f |w\rangle = -|w\rangle$$

$$U_f |x\rangle = |x\rangle , x \neq w$$



$$U_f |s\rangle = (I - 2|w\rangle\langle w|)|s\rangle$$

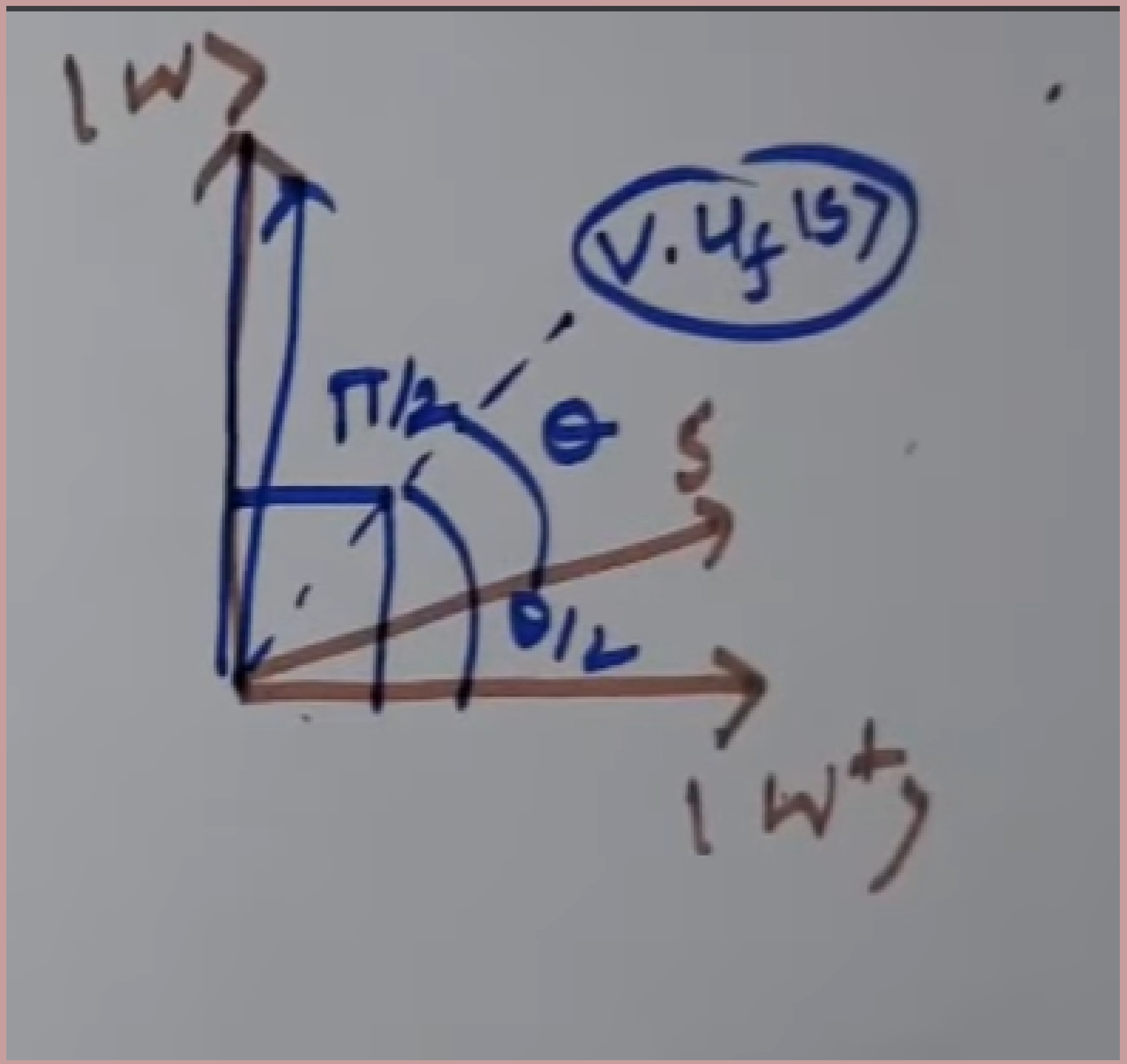


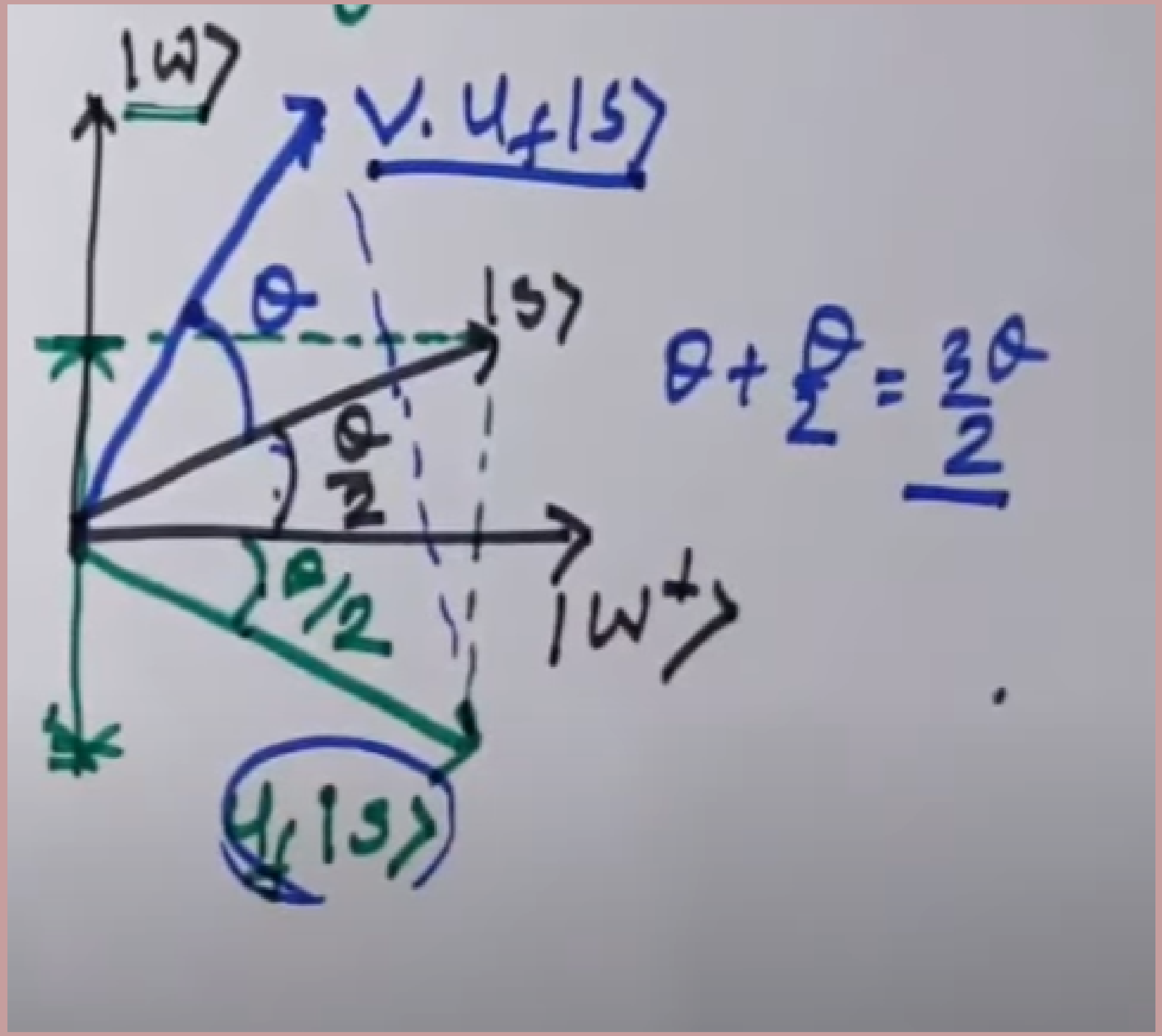


|0⟩
|0⟩
H
H
U_f
H
H
U_{f0}
H
H
V
|s⟩
$U_f \cdot |s\rangle$
$V \cdot U_f |s\rangle$

Step 3: inversion → the mean

$$V = (2|s\rangle\langle s| - I)$$





Time Complexity: O(√N)

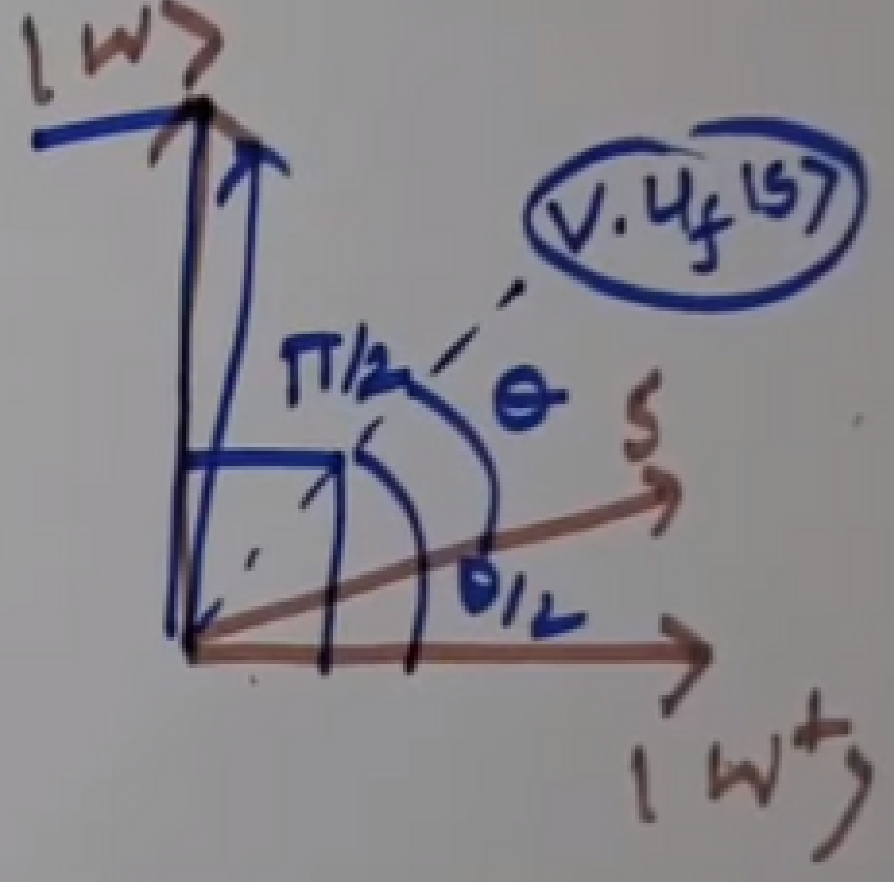
The time complexity of Grover's algorithm is determined by the number of iterations, which is roughly sqrt(N). Therefore, the overall time complexity of Grover's algorithm is O(sqrt(N)).

It's important to note that the time complexity analysis assumes ideal conditions and neglects the overhead of preparing the quantum state, applying gates, and performing measurements. In practice, the actual running time of Grover's algorithm is affected by various factors, such as the specific quantum hardware used, noise and errors in quantum operations, and the efficiency of the implementation.

$(V \cdot U_f)^n |s\rangle$

$|w\rangle$

$\theta = 2\arcsin\left(\frac{1}{\sqrt{2^n}}\right)$



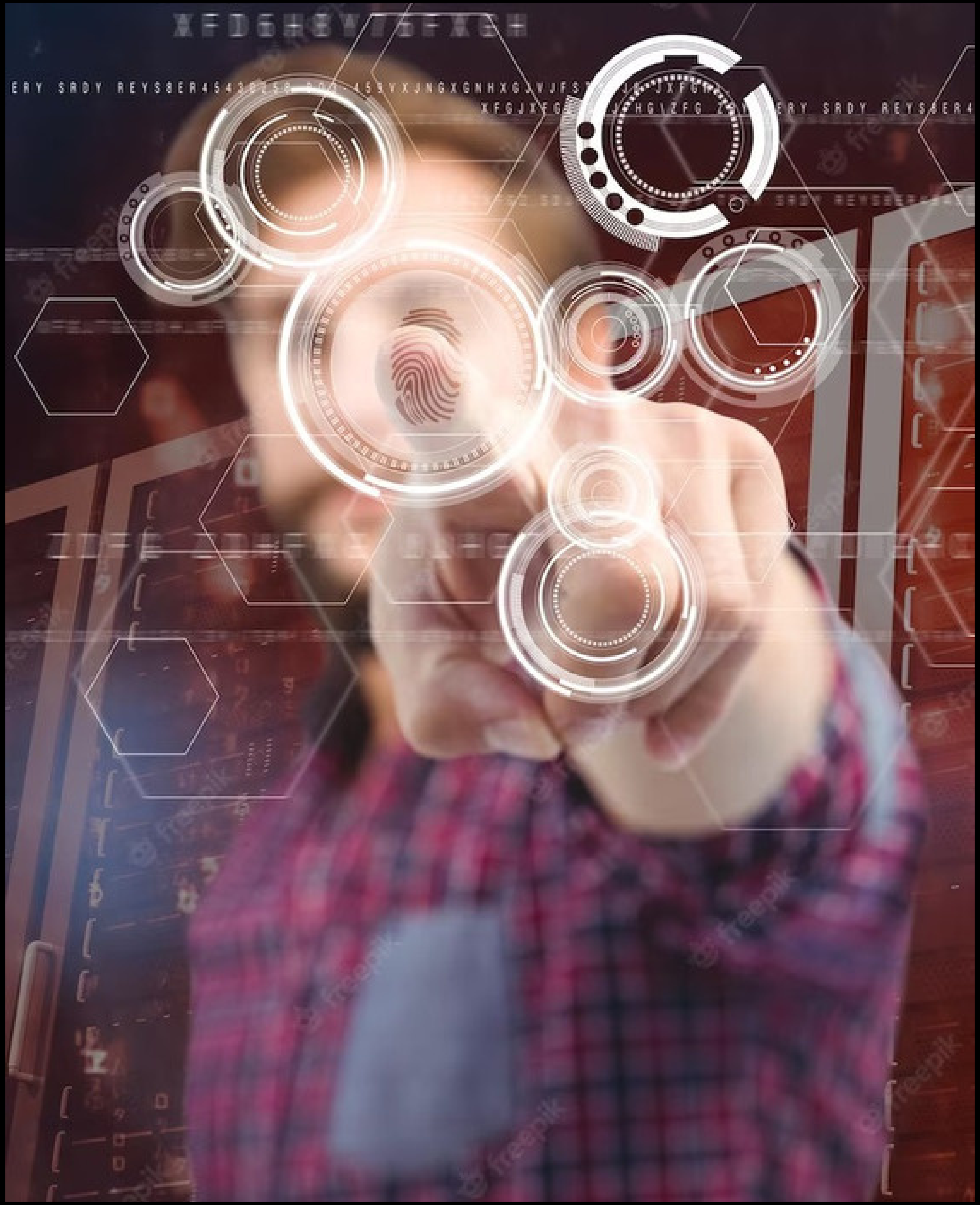
$$r\theta + \frac{\theta}{2} = \frac{\pi}{2}$$

$2^n = N$

$\sin\theta \approx \theta$

$$r = \frac{\pi}{2\theta} - \frac{1}{2}$$

$$= \frac{\pi}{2 \cdot 2\frac{1}{\sqrt{2^n}}} - \frac{1}{2} \approx \frac{\pi}{4}\sqrt{2^n} \approx O(\sqrt{N})$$



Applications of Grover's Algorithm

Grover's algorithm has wide-ranging applications, including **database search**, **cryptanalysis**, **machine learning**, and **optimization**. It can efficiently solve unstructured search problems, improve **database queries**, break **symmetric ciphers** faster, aid in **pattern recognition**, and optimize **combinatorial problems**. The potential impact of Grover's algorithm is immense, shaping the future of various fields.

Limitations and Challenges

While Grover's algorithm is a powerful tool, it also has limitations. It cannot outperform classical algorithms in all scenarios, as its speedup is limited to **search-based problems**. Additionally, implementing Grover's algorithm requires overcoming significant **technological challenges**, such as **quantum error correction** and **scalability**. Despite these obstacles, ongoing research and advancements are addressing these limitations.