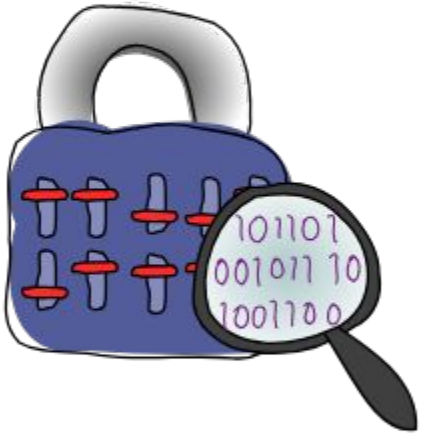
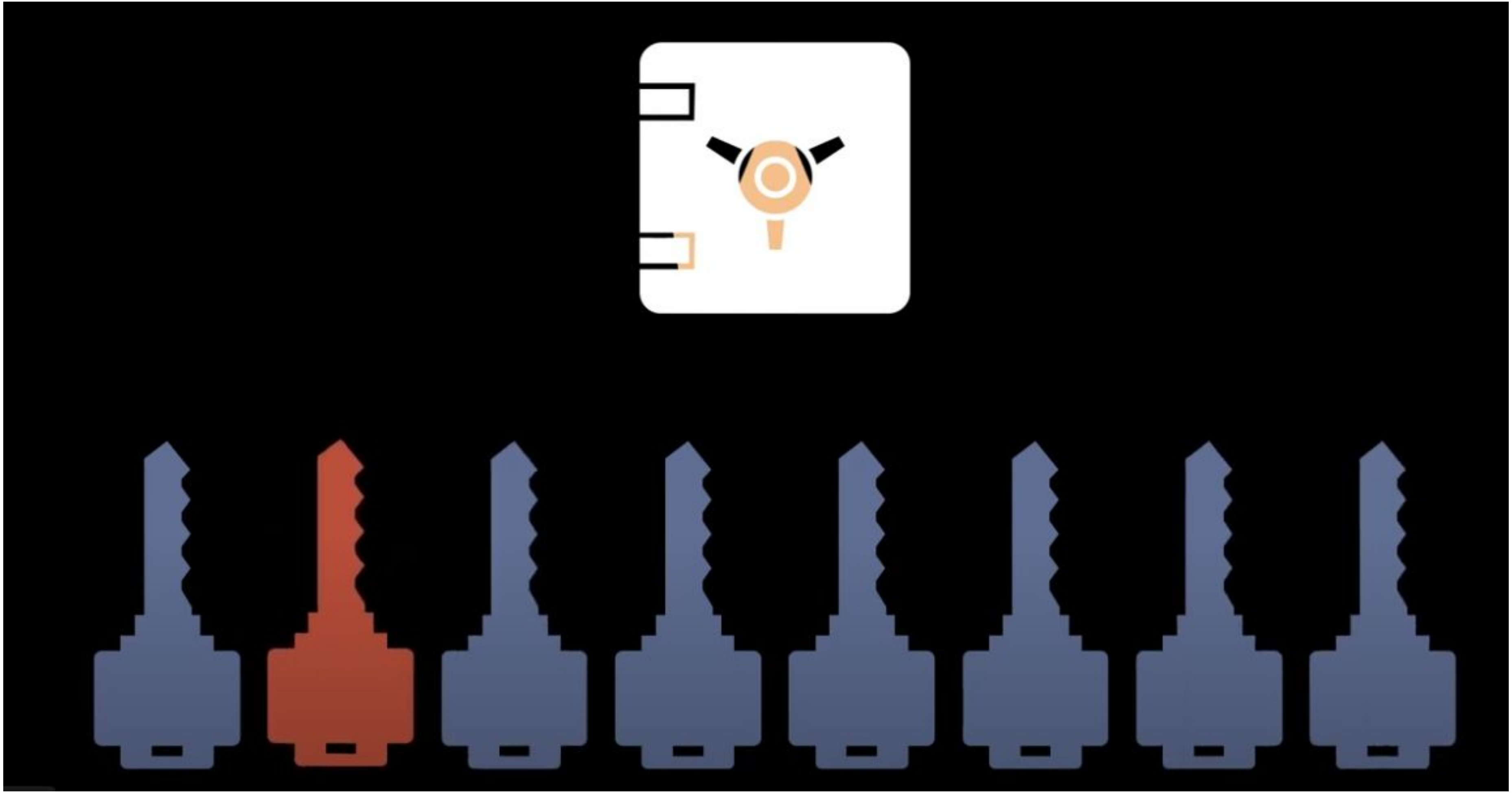
This is CS4048!

GCR:wzj3vua

Grover Algorithm





Grover Algorithm

The power of quantum parallelism, in specific, making queries in parallel.

Consider a function that is always equal to 0 except for a single value u. How are we going to find u?

$$f(x) = \begin{cases} 0 & \text{if } x \neq u \\ 1 & \text{if } x = u \end{cases}$$

Grover Algorithm

One of the many advantages a quantum computer has over a classical computer is its superior speed searching databases.

Grover's algorithm demonstrates this capability.

This algorithm can speed up an unstructured search problem quadratically,it can serve as a subroutine to obtain quadratic run time improvements for a variety of other algorithms.

This is called the amplitude amplification trick.

Grover Algorithm

Grover's algorithm consists of three main algorithms steps:

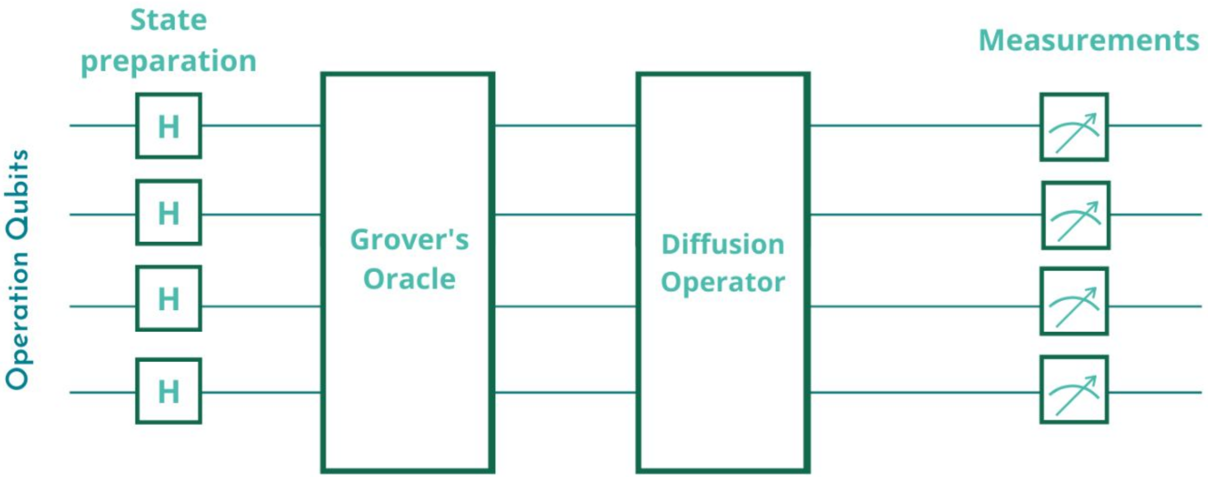
- State preparation
- Oracle
- Diffusion operator

Grover Algorithm

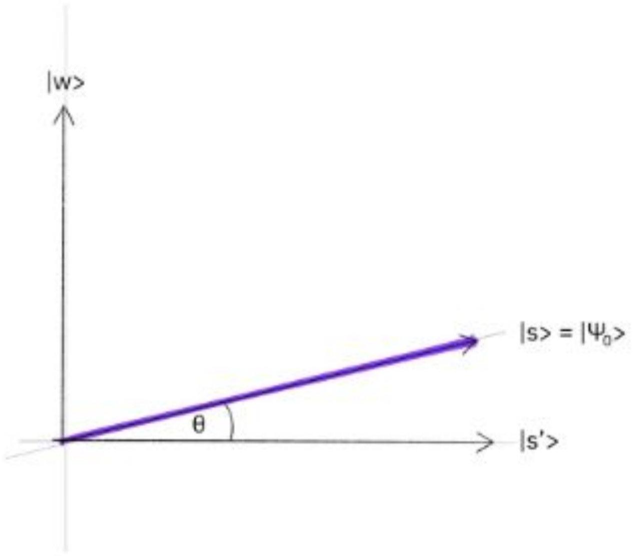
The state preparation is where we create the search space, which is all possible cases the answer could take.

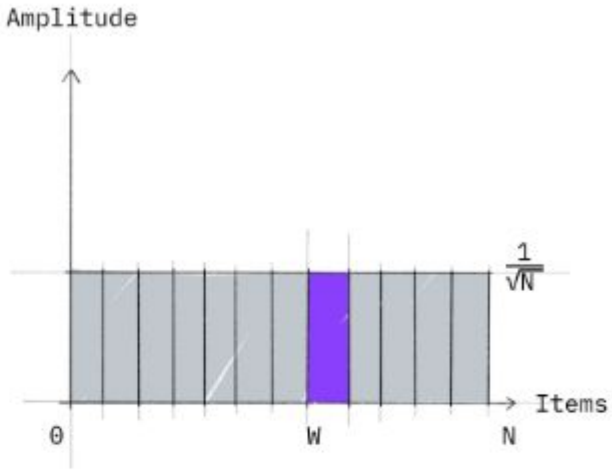
The oracle is what marks the correct answer, or answers we are looking for.

The diffusion operator magnifies these answers so they can stand out and be measured at the end of the algorithm.


State preparation
Measurements
Operation Qubits
H
H
H
H
Grover's Oracle
Diffusion Operator

Step 1: The amplitude amplification procedure starts out in the uniform superposition $|s\rangle$, which is easily constructed from $|s\rangle = H^{\otimes n}|0\rangle^n$ or using another symmetric entangled states.





The left graphic corresponds to the two-dimensional plane spanned by perpendicular vectors $|w\rangle$ and $|s'\rangle$ which allows to express the initial state as $|s\rangle = \sin\theta|w\rangle + \cos\theta|s'\rangle$, where $\theta = \arcsin\langle s|w\rangle = \arcsin\frac{1}{\sqrt{N}}$. The right graphic is a bar graph of the amplitudes of the state $|s\rangle$.

Grover Algorithm

$$U_\omega|x\rangle = \begin{cases} |x\rangle & \text{if } x \neq \omega \\ -|x\rangle & \text{if } x = \omega \end{cases}$$

$$U_\omega = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{array}{l} \\ \\ \\ \\ \\ \leftarrow \omega = 101 \\ \\ \\ \end{array}$$

Grover Algorithm

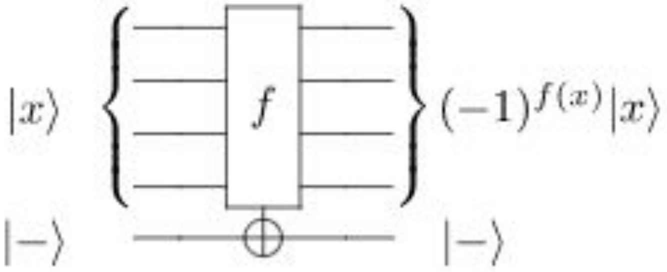
$$U_\omega|x\rangle = (-1)^{f(x)}|x\rangle$$

$$U_\omega = \begin{bmatrix} (-1)^{f(0)} & 0 & \cdots & 0 \\ 0 & (-1)^{f(1)} & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & (-1)^{f(2^n-1)} \end{bmatrix}$$

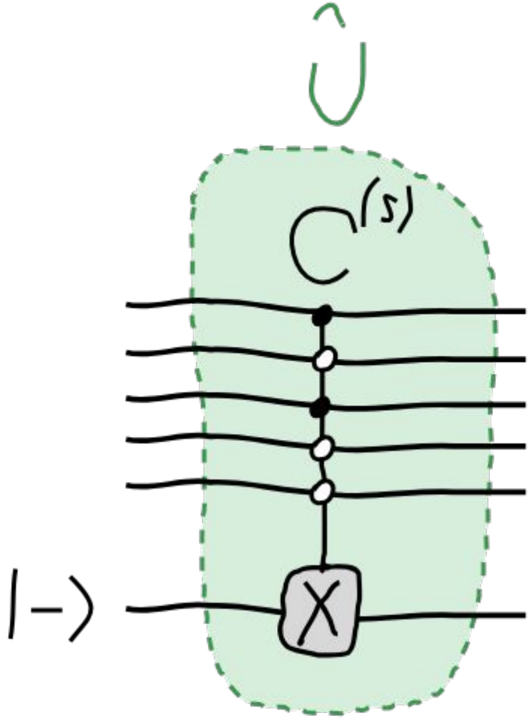
Grover Algorithm

If we initialize the 'output' qubit in the state |->, the phase kickback effect turns this into a Grover oracle (similar to the workings of the Deutsch-Jozsa oracle):

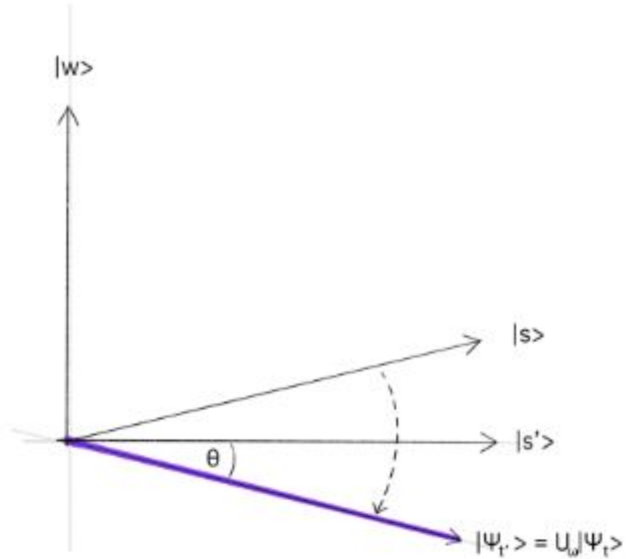
We then ignore the auxiliary qubit.

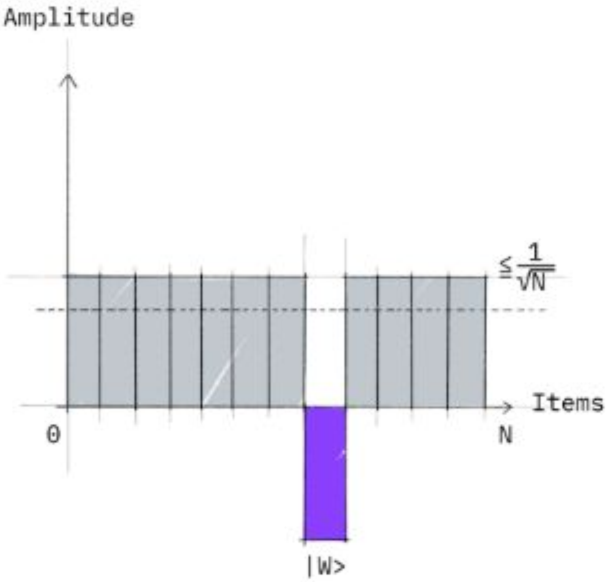


Oracle



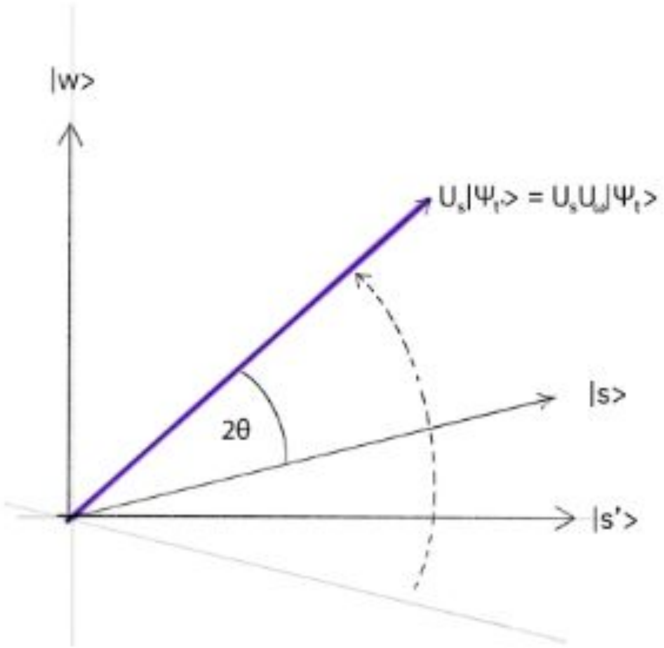
Step 2: We apply the oracle reflection U_f to the state $|s\rangle$.

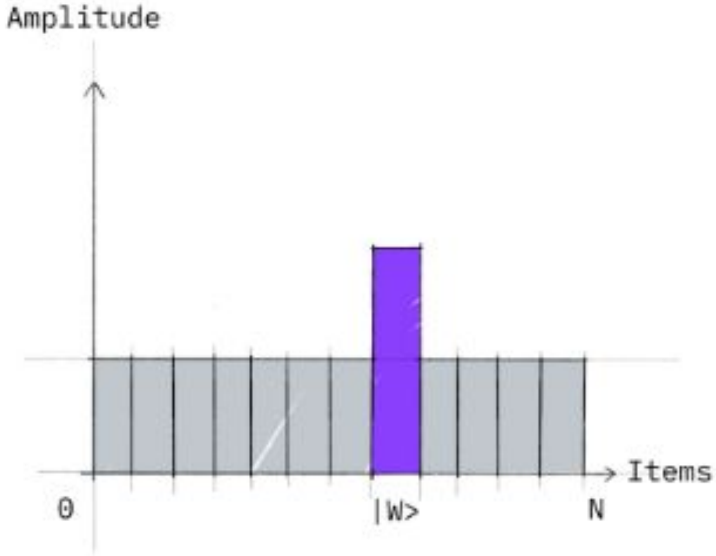


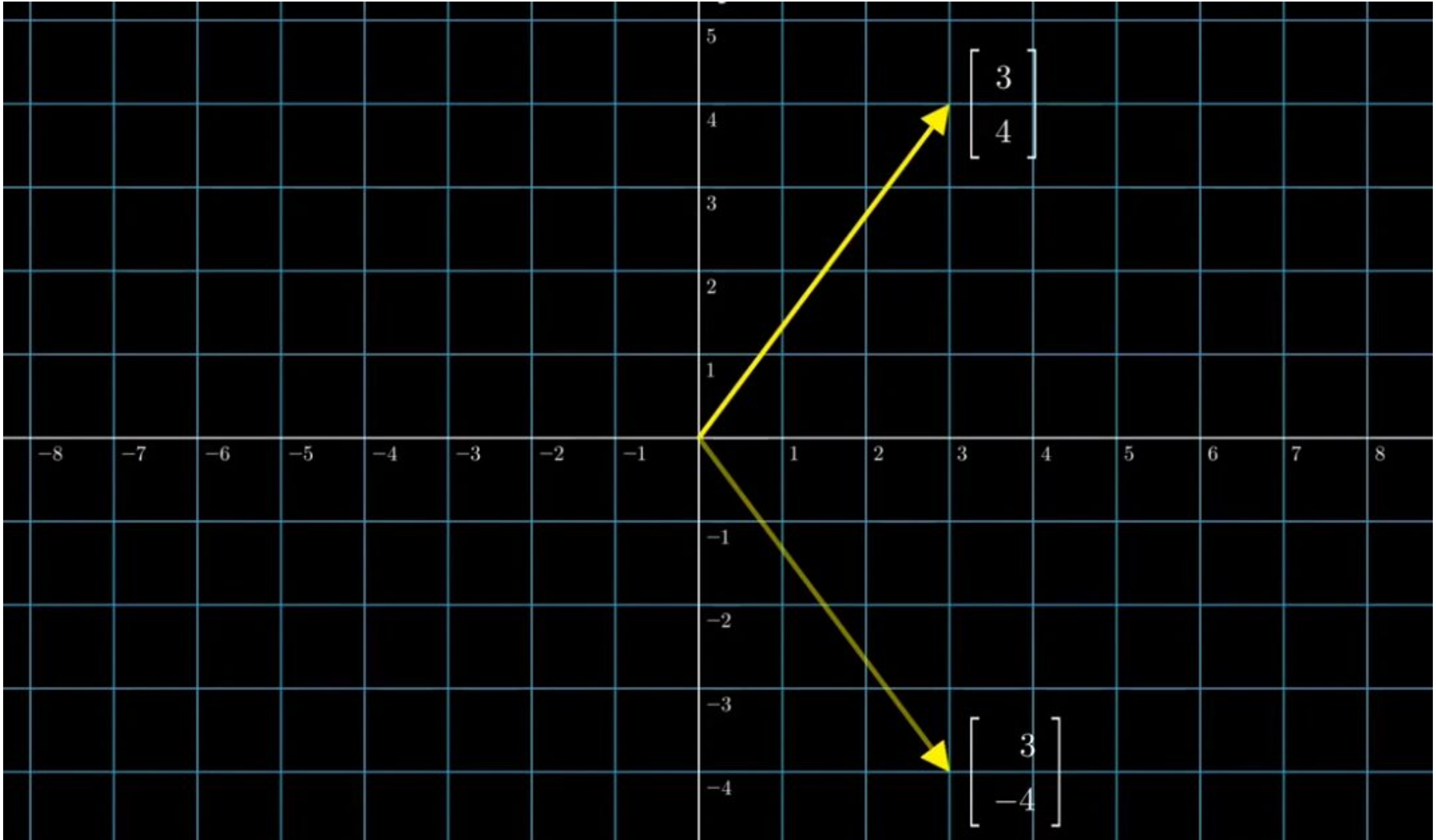


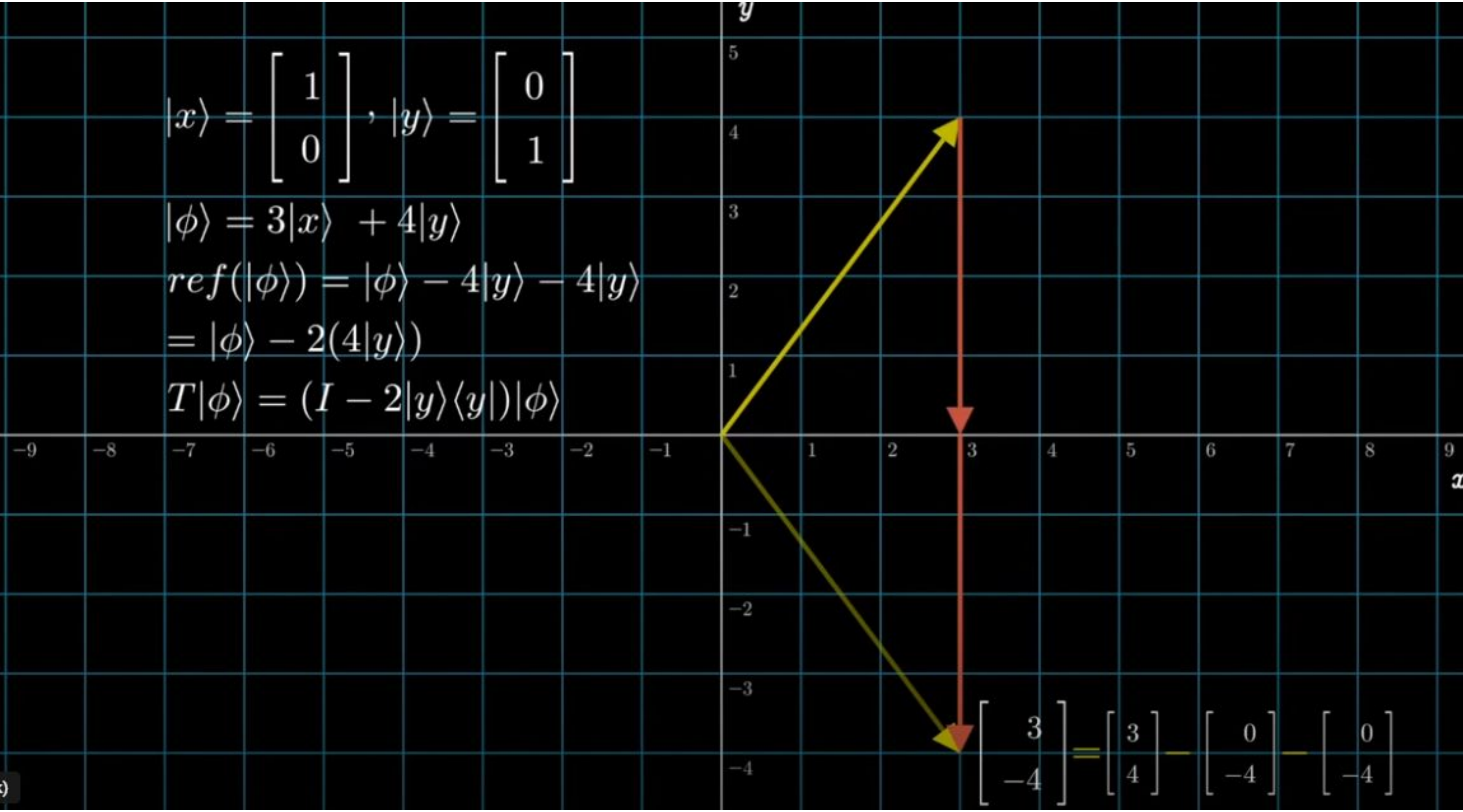
Geometrically this corresponds to a reflection of the state $|s\rangle$ about $|s'\rangle$. This transformation means that the amplitude in front of the $|w\rangle$ state becomes negative, which in turn means that the average amplitude (indicated by a dashed line) has been lowered.

Step 3: We now apply an additional reflection (U_s) about the state $|s\rangle$: $U_s = 2|s\rangle\langle s| - \mathbb{1}$. This transformation maps the state to $U_s U_f |s\rangle$ and completes the transformation.

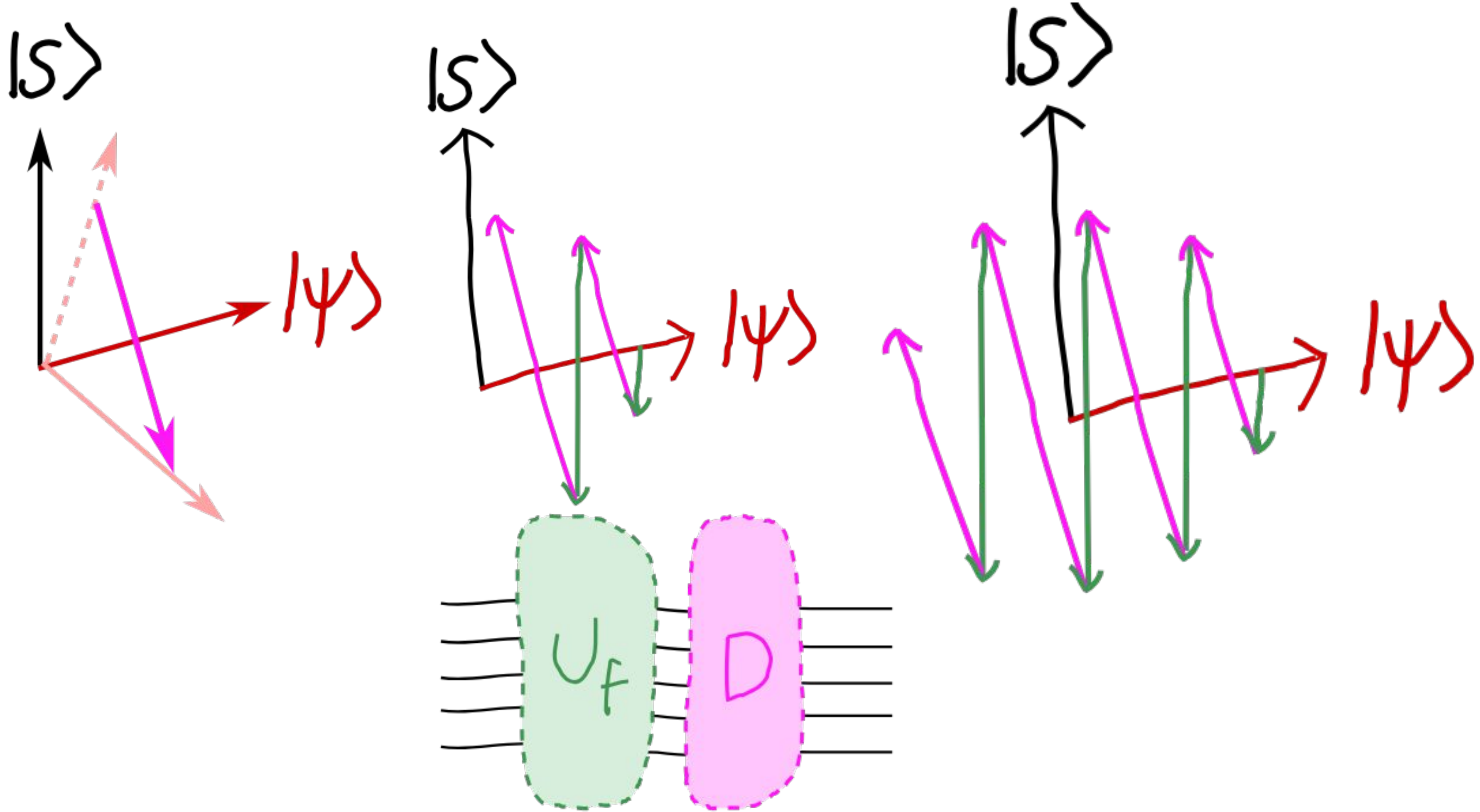





$\begin{bmatrix} 3 \\ 4 \end{bmatrix}$
$\begin{bmatrix} 3 \\ -4 \end{bmatrix}$
−8 −7 −6 −5 −4 −3 −2 −1 1 2 3 4 5 6 7 8
5 4 3 2 1 −1 −2 −3 −4



$|x\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |y\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$
$|\phi\rangle = 3|x\rangle + 4|y\rangle$
$ref(|\phi\rangle) = |\phi\rangle - 4|y\rangle - 4|y\rangle$
$= |\phi\rangle - 2(4|y\rangle)$
$T|\phi\rangle = (I - 2|y\rangle\langle y|)|\phi\rangle$
$\begin{bmatrix} 3 \\ -4 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} - \begin{bmatrix} 0 \\ -4 \end{bmatrix} - \begin{bmatrix} 0 \\ -4 \end{bmatrix}$

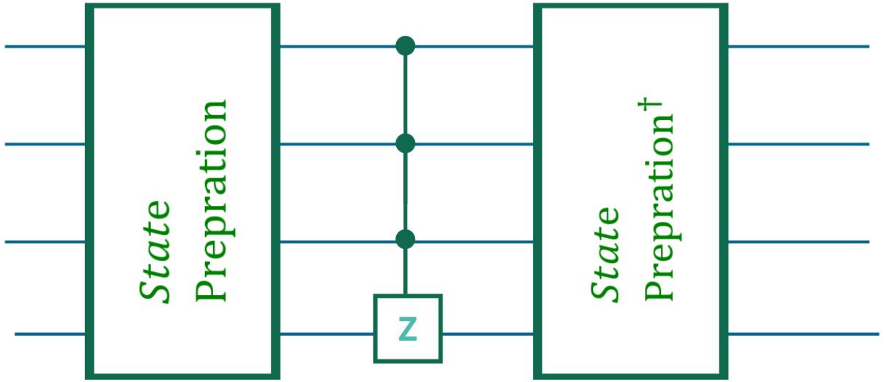


|s⟩
|ψ⟩
|s⟩
|ψ⟩
|s⟩
|ψ⟩
U_f
D

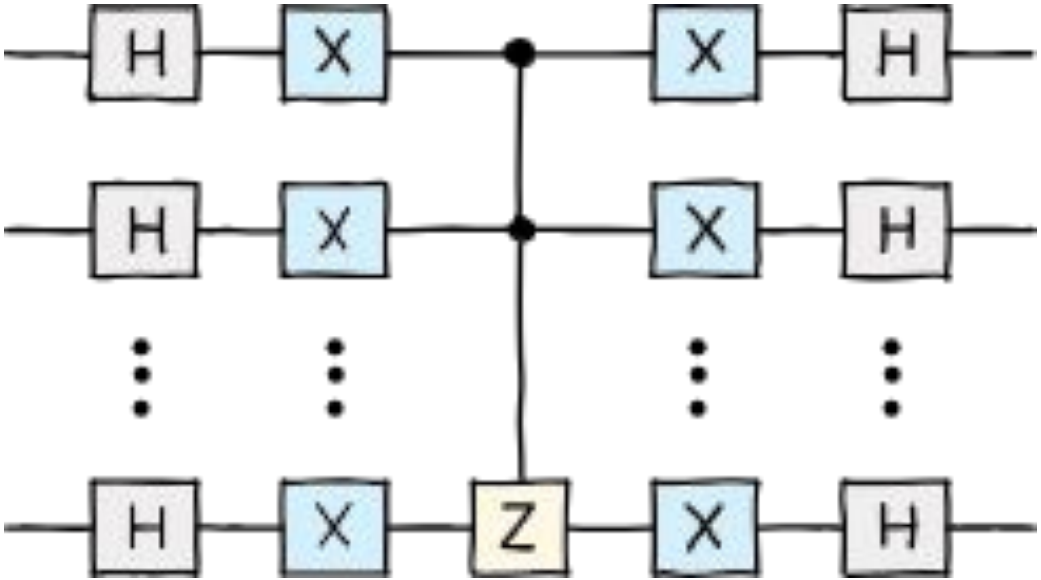
Grover Algorithm - Diffusion operator

After the oracle has marked the correct answer by making it -ve, the last step of Grover's algorithm is the diffusion operator.

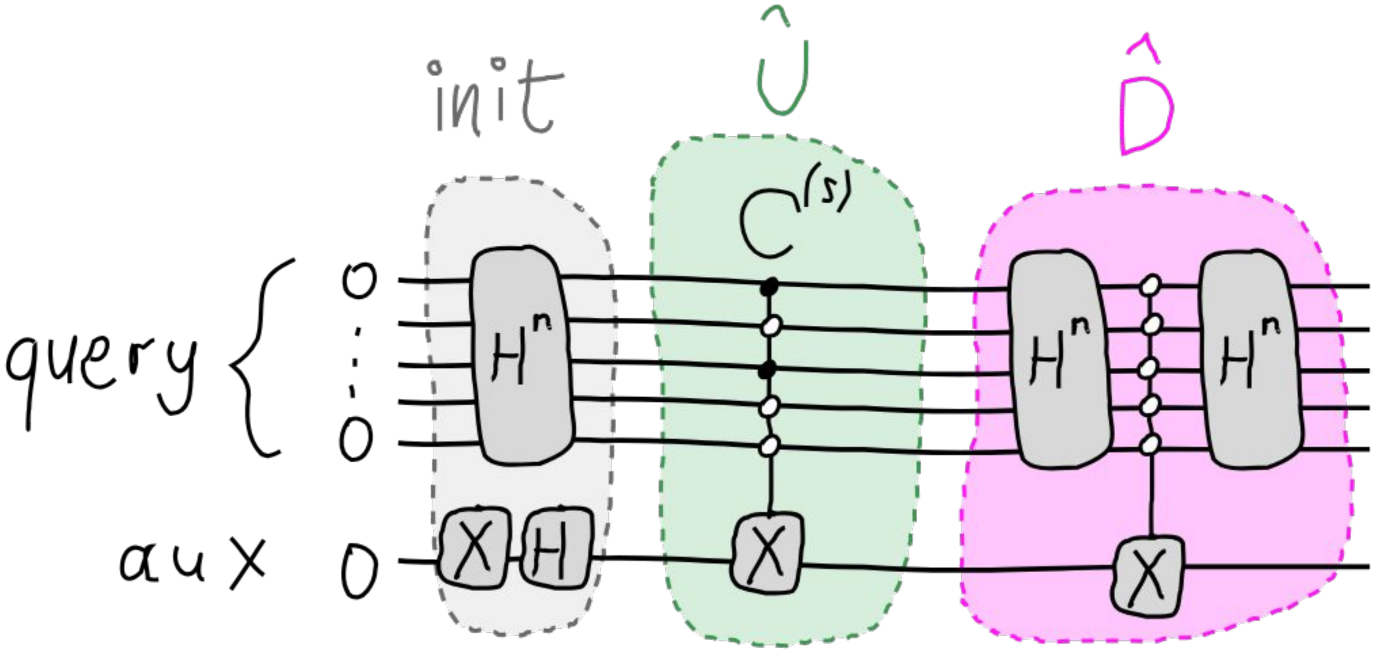
The construction of the diffusion operator depends on what we decide to use to prepare our initial states. Generally, the diffusion operator has the following construction.



Diffusion Operator

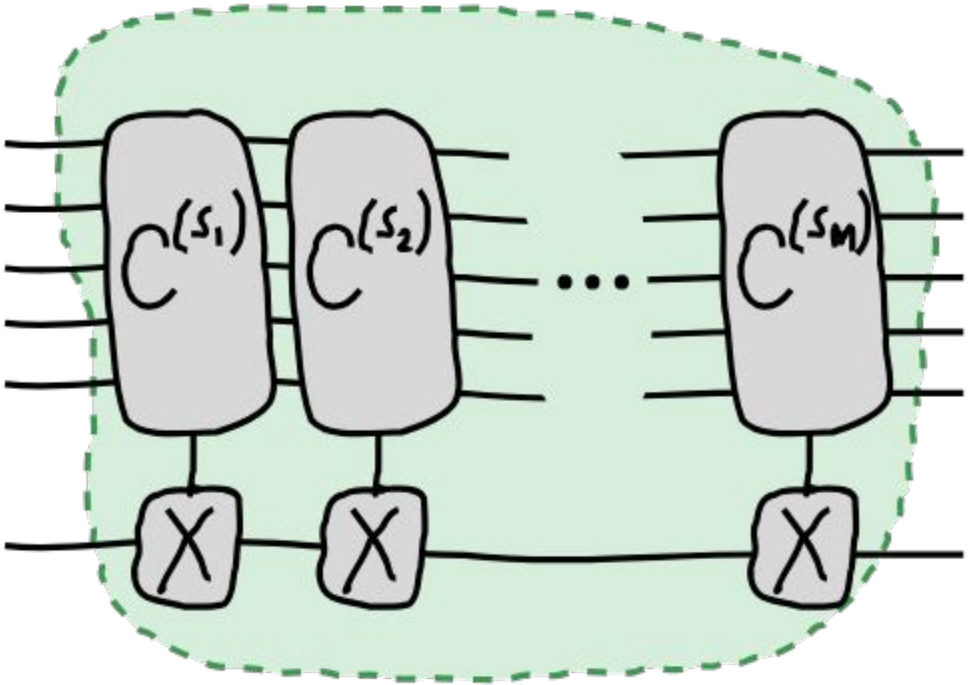


Grover Algorithm

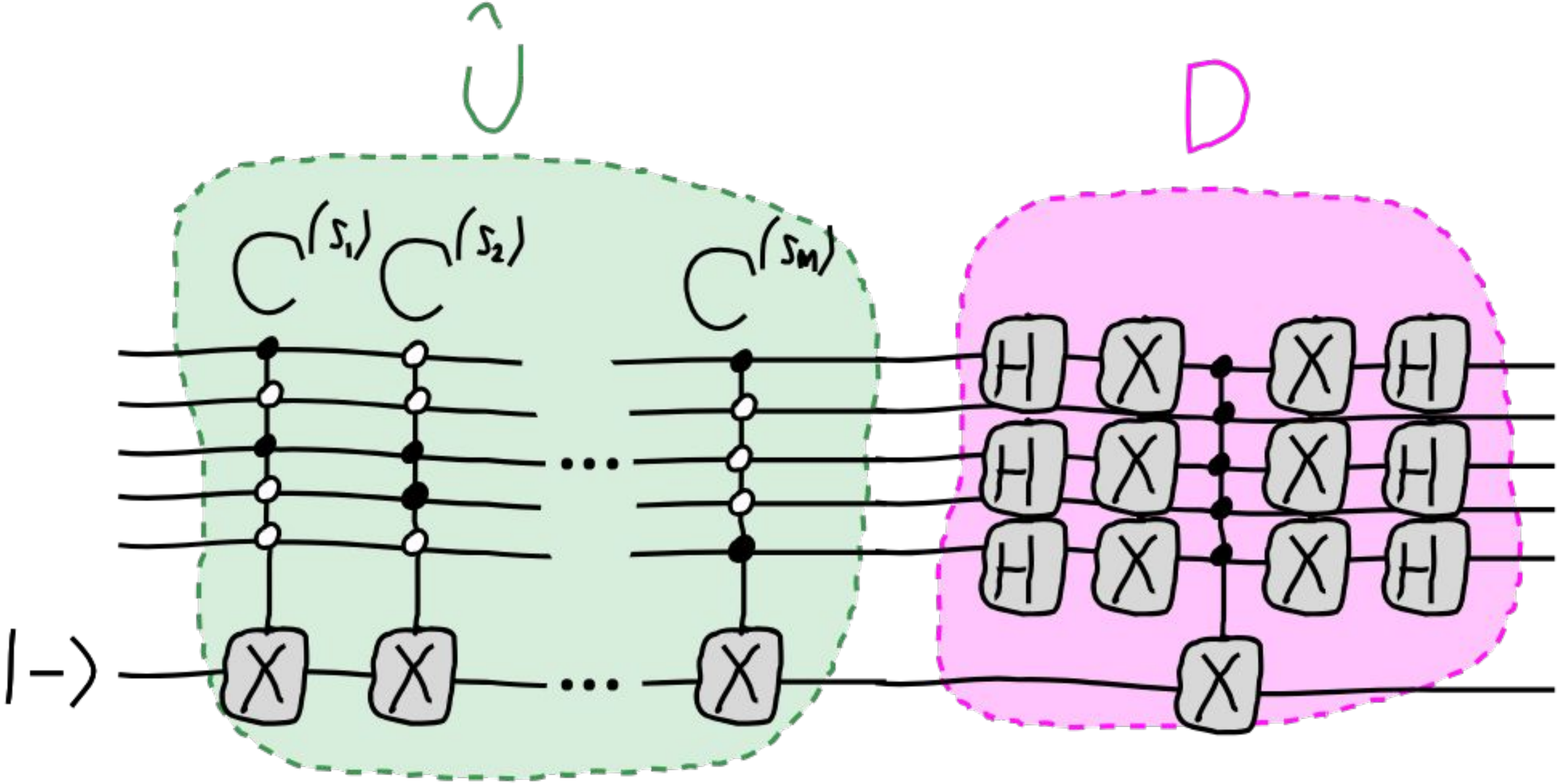


Multi Solution Oracle

$\hat{U}$



Multi Solution Oracle



Grover Algorithm

A classical algorithm requires N queries to find u in the worst case. But Grover's algorithm can complete the task in root N.

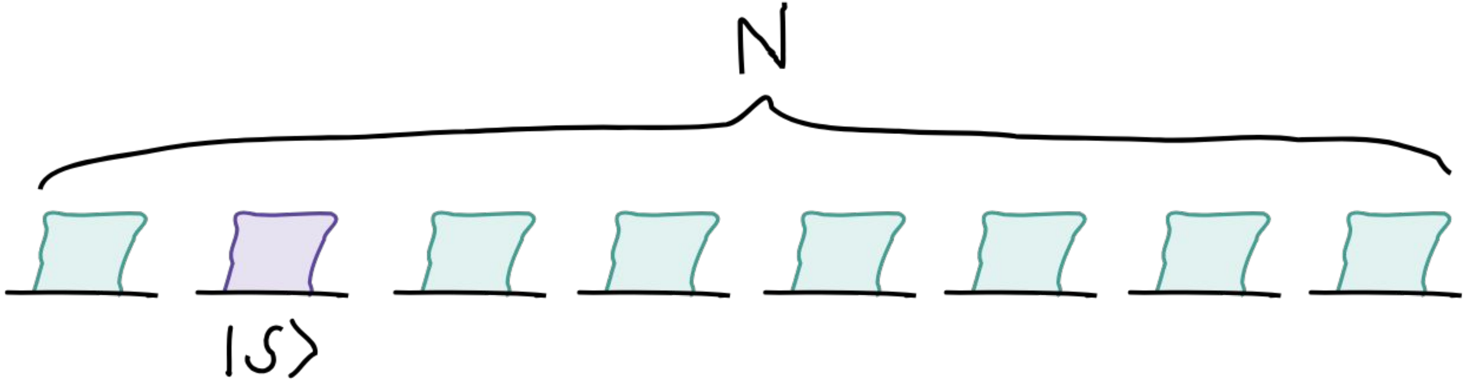
$$O(N) \quad \text{v.s} \quad O(\sqrt{N})$$

classical computing

quantum computing

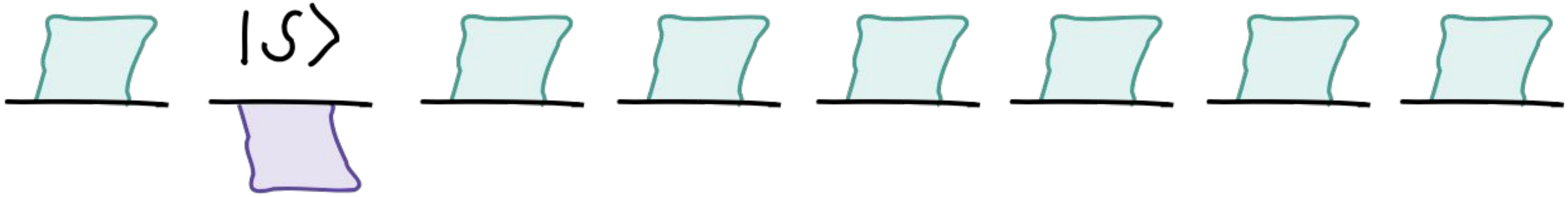
Grover Algorithm

Consider N orthogonal states. The second state is the solution. We want to somehow increase its amplitude, so we're more likely to observe it when we take a measurement.



Grover Algorithm

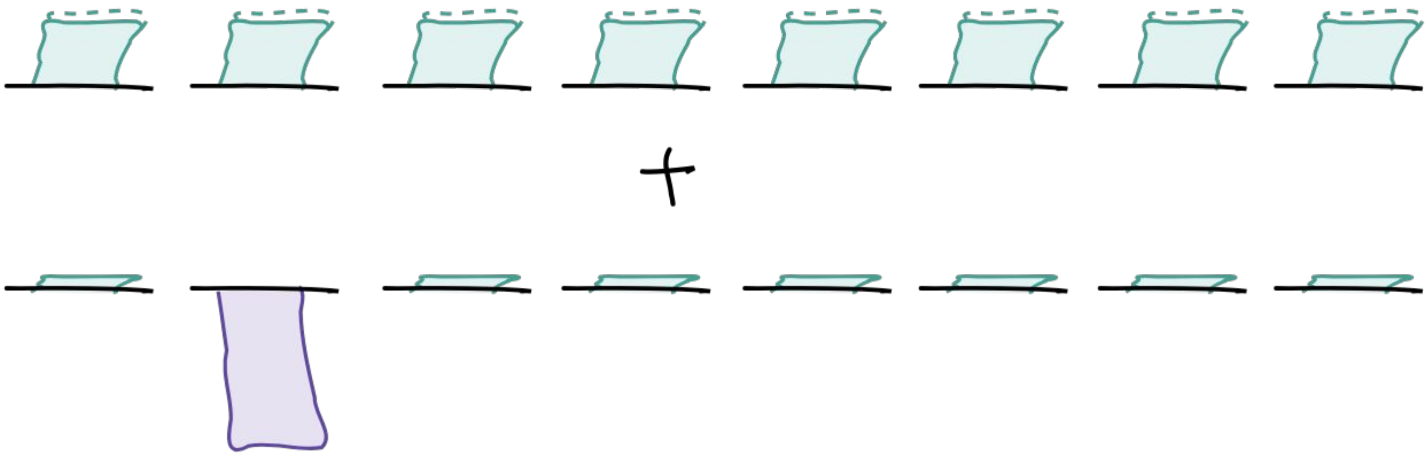
Our first step, as usual, is to apply the oracle, which flips the sign of the amplitude for :



Grover Algorithm

Applying the oracle again will only undo the phase flip.

Although the amplitude of is equal in size to the other states, it has a different phase.



https://github.com/Qiskit/textbook/blob/main/notebooks/intro/grover-intro.ipynb