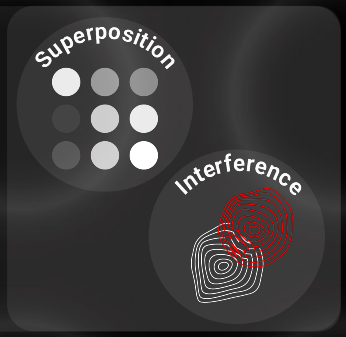
GROVER'S ALGORITHM

Grover's algorithm is a quantum search algorithm that uses two properties of quantum objects to accelerate the identification of a particular item (or items) in a set.

Specifically, it uses **superposition** to take into consideration all possible combinations in which our set can be mixed, or to take in consideration every item of the set, all at once.

It then uses **interference** to amplify the probability of finding the particular item we are searching for, and to reduce the likelihood of getting the wrong one.



IMPORTANCE AND USABILITY

- Quantum computers are not meant to replace classical ones, but rather, use their properties to achieve things impossible for classical computers, or that would require a longer amount of time (*like this algorithm*).
- When items on a dataset are unordered, classical computers take O(N) times to find a specific item because they have to check every item one by one. Grover's algorithm has a **quadratic advantage**, taking O(√N) times.
- Although the algorithm *uses* **superposition** to take into consideration in one (1) computation all possible items, this doesn't mean that it will only take one run to find the item, the algorithm requires a specific number of runs to **augment** the probability of finding it.
- By adding more qubits to our system, we can represent more items (or combinations).
- Current QC have a limited amount of qubits due to the **elevated cost of the hardware** to keep the qubits isolated from the outside world and avoid interference with any other particles
- Another obstacle is the qubits **decoherence** , which causes qubits to maintain their state ONLY for a limited amount of time, most of them less than a second.
- Practical tests using the algorithm had less than 10 items on the dataset and the protocol hasn't been used on real life problems yet.
- Classical computers still take the lead on searching through ordered datasets and have many more bits for the unordered ones.

That does not mean it will never be used. Just think of the algorithms of Ada Lovelace, or the Alan Turing tests, and how it took years to take to practical use, nonetheless, without a first step, there's no walking.

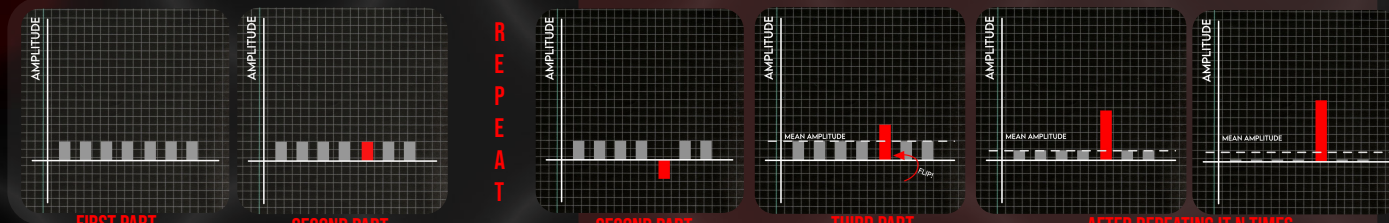
USE CASE - FORENSICS PERPETRATOR IDENTIFICATION SYSTEM

We could either use the Grover's algorithm for **combinatorial** problems, or to **search a database**.
In this case, we will choose to apply the algorithm to a database search problem. We will simulate the use of it creating a "**Forensics perpetrator identification**" system.

In some cases, victims of a crime like robbery are asked to provide details of the perpetrator, so a specialized artist can create a sketch of the subject's face.

Instead of "creating" a person from the characteristics given, we could "search" for a real one from a database of people, where facial or body features have been digitized.

CIRCUIT

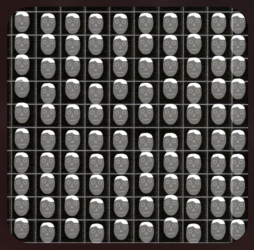


QUBIT X QUBIT

STEPS

1.Define the size of the problem.

We have to encode our problem into bits. We could say that if we have 7 qubits available, we then have 128 possible combinations (2^{qubits}), or in our case, 128 records of people in our database, each one of them with different characteristics.



2. Create the oracle.

Our oracle is a function or (black box) that suffices f(x) = 1. In our case, depending on the person we are looking for, it could look like the following logical expression:

$$Male \wedge \sim(Hair) \wedge (>65 years \wedge <75 years) \wedge (RoundedEyebrows \wedge ThickEyebrows)$$

The above would mean that we are looking for a **"male person, with hair (not bald), and who is between 65 and 75 years old, and has rounded and thick eyebrows". If everything suffices, it would have to return a boolean "True", which we can interpret as 1 so that f(x) = 1.**

3. Create the circuit.

Most Grover's algorithm circuits consist of the same gates being applied, quantum gates are operations made to the qubits to changes their state.
FIRST PART of the circuit is applying an H gate to all qubits (all items on the database), to put them on equal superposition.

SECOND PART is applying our "Oracle" to the circuit so that it marks the qubit that meets our perpetrator description (the formula on Step 2), and then flips its phase.

THIRD PART consists of applying the so called "Diffusion operator" which will flip all qubits in relation to the mean amplitude across them all, this operation will reduce the amplitude of the incorrect ones and increase it on our correct marked qubit. This increases the probability of measuring that qubits in comparison to the others.

4. Repeat

We have to repeat the "Flip Phase" and "Diffusion Operator" parts $\sqrt{N}$ times due to the probabilistic nature of quantum computing (in our case with N= 128, that's around 11 times);

- While it's true that qubits can be in a superposition of 1 and 0 at the same time, when we measure them (check them), they will be in either 1 or 0, not both, however, the probability of measure one or the other can be modified to our favor, **and that's what we are repeating these parts.**

5. Measure!

Finally, we take a measurement of our circuit and, we have a high enough probability of finding our qubit in the state that represents our perpetrator.