Grover's Algorithm Tutorial/Experimentation

Rahul Goel

November 2020

Note - It has been assumed that $N = 2^n$ throughout this sheet.

1 The Problem - Needle in a Haystack

The problem that Grover's Algorithm solves is the unstructured search problem. In simple words, among N given entities, our task is to find one unique entity that satisfies a particular property.

In most cases, the situation is random and the only way a classical computer knows how to solve it is by brute force. This can take $O(N/2)$ on average and up to $O(N)$ time in the worst case.

The problem of unstructured search is considered to be a hard problem for a classical computer as N gets bigger in size. Grover's Algorithm provides a quadratic speedup to this and reduces the number of queries (time taken) to $O(\sqrt{N})$. In fact, it has been proven that for a quantum computer, this is the best possible time complexity.

Note - The problem gets easier and the complexity gets reduced by a constant factor when the entities that satisfy the property are more in number.

2 How the Algorithm works

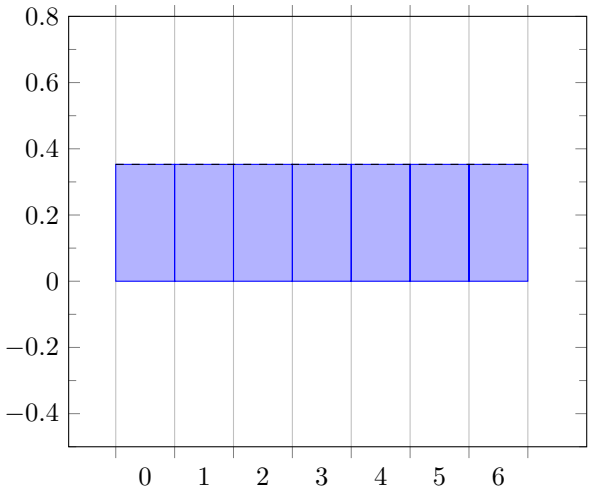
2.1 Intution 1

Initially, we start with n qubits in such a way that in their superposition, all the N states have equal amplitude (each has an equal probability of popping up when we measure the state).

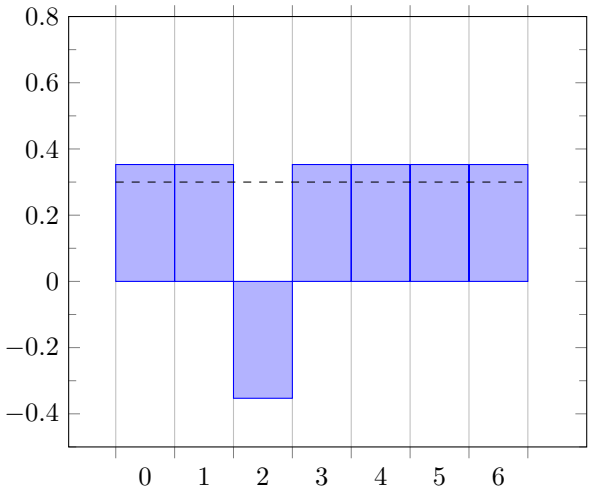
$$|\psi_{initial}\rangle = \sum_{0}^{N-1} \alpha_i |i\rangle \text{ where } \alpha_i = \frac{1}{\sqrt{N}}$$

Let $|j\rangle$ be our needle in the haystack. Now, by means of certain quantum gates, the amplitude α_j is first negated i.e. it becomes $-\alpha_j$. And then every single α_i is flipped about the mean of all amplitudes. As a result, α_j stands out more than every other α_i.

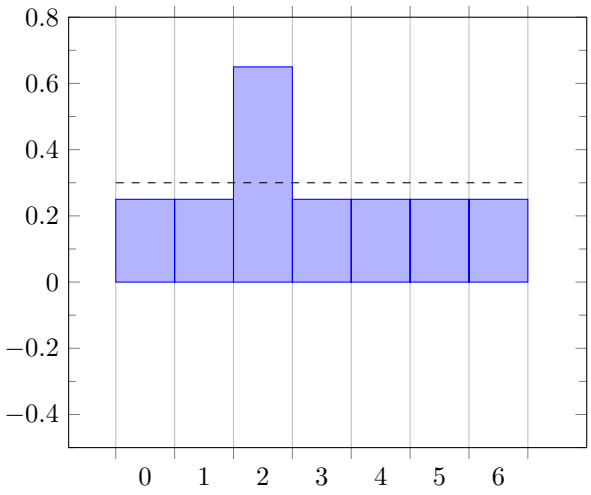
This has been explained by means of the following diagrams (Not to scale). Initially the amplitudes of all the states are equal and the sum of their squares is equal to 1.



Then a quantum circuit is applied to the state which flips the amplitude of the desired state.



Now the inversion of one of the amplitudes results in a slight decrement of the mean of all the amplitudes. If we now flip the amplitudes about their mean, we land up in the following state.



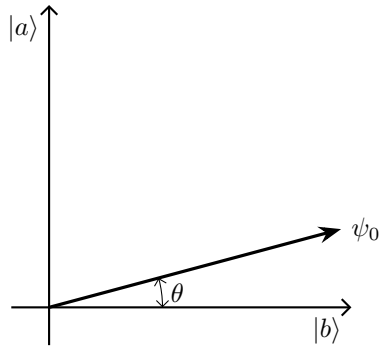
The amplitude of the needle in the haystack has increased and hence if we measure the state now, the chance of getting the correct value is more than it was previously. However, the sum of probabilities of getting other states is still greater than the probability of getting the answer.

Hence, we repeat the negation and inversion steps a number of times so that when we measure the state, we can be pretty sure that the outcome is the answer (the desirable state).

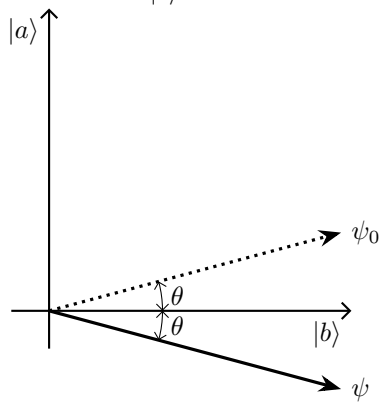
Initially $\alpha_j = \frac{1}{\sqrt{N}}$. As we run iterations of the algorithm, α_j increases and α_i, $\forall\ i \neq j$, decreases, all in such a way that $\sum_0^{N-1} {\alpha_i}^2 = 1$. If we want the probability of getting $|j\rangle$ to be at least half, we need to pump up the magnitude α_j to $\frac{1}{\sqrt{2}}$. At the point in time when $\alpha_j = \frac{1}{\sqrt{2}}$, $\alpha_i \approx \frac{1}{\sqrt{2N}}\ \forall\ i \neq j$. Because the amplitude of the other states decreases continuously, we can say that throughout our iterations $\alpha_i \geq \frac{1}{\sqrt{2N}}$. Since in one iteration, the amplitude α_j increases by at least by $2\alpha_i$ $(i \neq j)$, the increment is at least $\sqrt{\frac{2}{N}}$. Therefore the number of iterations required is at least $\frac{1/\sqrt{2}}{\sqrt{2/N}} = \sqrt{N}$.
When the number of favourable states is M, the number of iterations comes out to be $\sqrt{\frac{N}{M}}$.

2.2 Intution 2

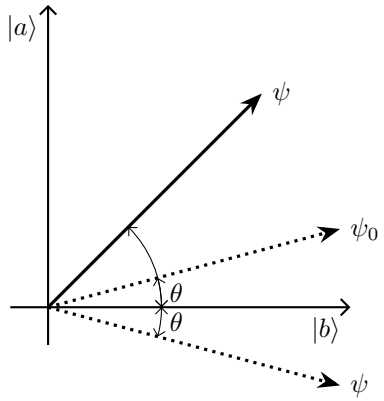
Let $|a\rangle$ represent our desirable state i.e. $|j\rangle$. And $|b\rangle$ represent the superposition of all the remaining states with equal amplitudes i.e. $|b\rangle = \sum_{i \neq j} \frac{1}{\sqrt{N-1}} |i\rangle$. It can be easily noticed that $|a\rangle$ and $|b\rangle$ are orthogonal to each other. Let ψ_0 be our initial state which is $\sum_0^{N-1} \frac{1}{\sqrt{N}} |i\rangle$. Now initially $|\psi_0\rangle$ lies really close to $|b\rangle$ and it is the task of Grover's Algorithm to move it closer to $|a\rangle$.



The amplitude negation step leads to reflection of $|\psi_0\rangle$ about $|b\rangle$. This is true because $|b\rangle$ represents the superposition of all the undesirable states of equal amplitude. So the projection of $|\psi_0\rangle$ on $|b\rangle$ will remain as it is while the projection of $|\psi_0\rangle$ perpendicular to $|b\rangle$ will get negated. This is nothing but reflection about $|b\rangle$.



Now, we need to do inversion about the mean which is equivalent to reflection about the the equal amplitude state $\sum_0^{N-1} \frac{1}{\sqrt{N}} |i\rangle$.



So, after one iteration we have moved closer to the required state. We need to carry these iterations certain number of times so that when we measure a state, it collapses to the desirable state with high probability.

Every time the inversion about mean is done using the equal amplitude state only. It happens to be our initial state too but this is true only for the first iteration and not for further iterations.

3 How to Implement the circuit?

Now that we know what to do, we have to question how to do it.

3.1 Negation of Amplitude

Our first task is to negate the amplitude of the desirable quantum state. The situation is like this that we have been given a quantum black box that outputs 1 for the desirable states and 0 for the undesirable states.

Insert diagram here.

In other words, given the superposition $|x\rangle |0\rangle$, the circuit spits out the superposition $|x\rangle |f(x)\rangle$. We know the quantum circuits are designed in such a way that the answer register gets XORed with the output $f(x)$. So if the input superposition is $|x\rangle |y\rangle$, then the output superposition will be $|x\rangle |y \oplus f(x)\rangle$.

With this knowledge, we can apply neat little hack and trick our black box into doing the task for us. We will initialise the answer register to $|-\rangle$. Since

$$
\begin{aligned}
|-\rangle &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \\
|- \oplus f(x)\rangle &= \frac{1}{\sqrt{2}}\left(|0 \oplus f(x)\rangle - |0 \oplus f(x)\rangle\right) \\
|- \oplus f(x)\rangle &= \frac{1}{\sqrt{2}}\left(|f(x)\rangle - \left|\overline{f(x)}\right\rangle\right) \\
|- \oplus f(x)\rangle &= \frac{1}{\sqrt{2}}(-1)^{f(x)}\left(|0\rangle - |1\rangle\right) \\
|- \oplus f(x)\rangle &= (-1)^{f(x)}|-\rangle
\end{aligned}
$$

Therefore, the entire transition will be

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

Now, since it is a superposition that is being looked at as a whole, the term $(-1)^{f(x)}$ can be associated with either $|x\rangle$ or $|-\rangle$. Putting $|x\rangle = \sum_0^{N-1}\alpha_i|i\rangle$, we get the output from the black box as $\sum_0^{N-1}\alpha_i|i\rangle(-1)^{f(x)}$. Now since $f(x) = 1$ for $|j\rangle$ and 0 for all others, the amplitude is negated only for $|j\rangle$. Thus we can do the amplitude negation easily.

3.2 Inversion about mean

Inversion about mean is the same as reflection about the uniform state i.e. reflection about the state $\sum_0^{N-1}\frac{1}{\sqrt{N}}$. Now for reflection about the uniform state, we can first map the uniform state to the all-zero state using a Hadamard Gate and then reflect about the all-zero state and then map the state back to the uniform state using another Hadamard Gate (Since Hadamard Gate is it's own inverse). So now we need a method to reflect about the all-zero state ($|00\ldots0\rangle$ or simply $|0\rangle$). Consider the following matrix :-

$$A = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & -1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & -1 \end{bmatrix}$$

This matrix maps negates all the states except for the state $|0\rangle$. This is the same as reflection about $|0\rangle$. This matrix is also a unitary matrix because $AA^\dagger = I$ i.e. the conjugate transpose is its inverse. So we can conclude that the sequence of matrices $H_N A H_N$ (where H_N stands for the unitary matrix corresponding to Hadamard Gate applied to n qubits.) does a reflection about the mean. And since all the matrices are unitary, they can be implemented using quantum gates.

So when we apply the operations of amplitude negation and inversion about mean repeatedly for around $\sqrt{N}$ times, the probability of getting the correct answer exceeds half and we are highly likely to get the current answer. We can perform this several times to be sure of the answer since the method is probabilistic.

4 Qiskit Code

Note - Please refer to the Jupyter Notebook for the output of the program and the explanatory comments and further experimentation.

Simulation Code

```
import matplotlib.pyplot as plt
import numpy as np

from qiskit import IBMQ, Aer, QuantumCircuit,
    ClassicalRegister, QuantumRegister, execute
from qiskit.providers.ibmq import least_busy
from qiskit.quantum_info import Statevector
from qiskit.visualization import plot_histogram
from qiskit.extensions import UnitaryGate
from math import sqrt

n = 5
N = 2 ** n

def initialize(qc, qubits):
    for q in qubits:
        qc.h(q)
    return qc

def get_oracle_matrix(N, values):
    oracle_matrix = np.identity(N)
    for value in values:
        oracle_matrix[value][value] = -1
    return oracle_matrix

def get_diffusion_matrix(N):
    diffusion_matrix = np.zeros((N, N), dtype = float)
    diffusion_matrix.fill(2 / N)
    diffusion_matrix -= np.identity(N)
    return diffusion_matrix
```

Simulation Code

```
oracle_matrix = get_oracle_matrix(N, values = [2])
oracle_unitary_gate = UnitaryGate(oracle_matrix)

diffusion_matrix = get_diffusion_matrix(N)
diffusion_unitary_gate = UnitaryGate(diffusion_matrix)

qc = QuantumCircuit(n)
qc = initialize(qc, [x for x in range(n)])

for i in range(5):
    qc.unitary(oracle_matrix, [x for x in range(n)])
    qc.unitary(diffusion_matrix, [x for x in range(n)])
qc.draw()

qc.measure_all()

qasm_simulator = Aer.get_backend('qasm_simulator')
shots = 1024
results = execute(qc, backend=qasm_simulator,
                  shots=shots).result()
answer = results.get_counts()
plot_histogram(answer, figsize = (15, 12))
```

5 Findings

While it was mostly studying stuff discovered by others, I did observe some things on my own that most probably have already been observed.

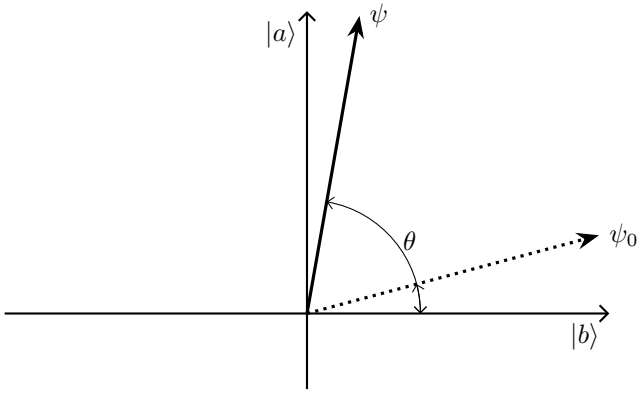
5.1 Going beyond $\sqrt{N}$ queries

In classical computers, it just doesn't hurt to go a few extra steps (since they don't harm the overall time complexity) just to be sure of our result. I tried this out on the implementation that I wrote and the results surprised me. After exceeding the required number of queries the amplitude of the desired state starts to decrease again and reaches a certain value and increases again and oscillates like this. Also, the period of this oscillation varies along with the number of qubits we are dealing with. When the number of qubits is small, the value decreases quite suddenly and even deviating by 1 query can alter the probability of the outcome easily. However, when the number of qubits is large this is smoothened out, and a query or two give or take doesn't affect the outcome by much.

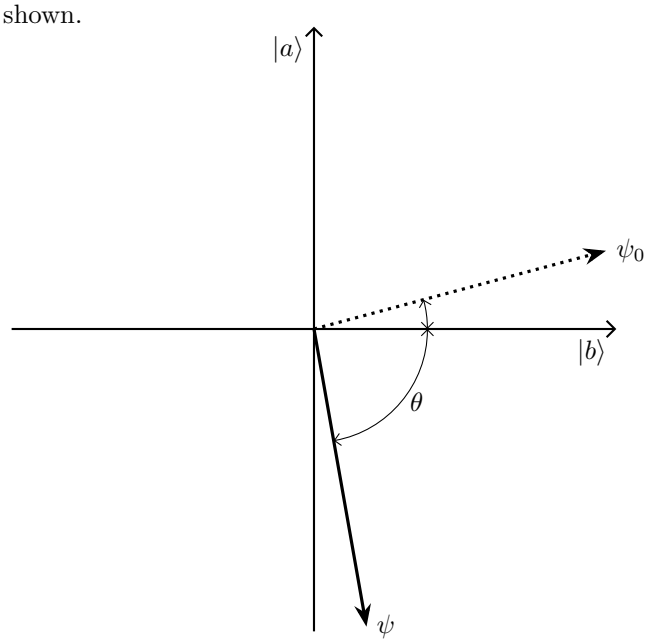
All in all, minimum $\sqrt{N}$ queries are required and should be sufficient to give the answer. One should not go beyond this limit. Rather, one should re-conduct

the experiment from the initial state to verify the answer.

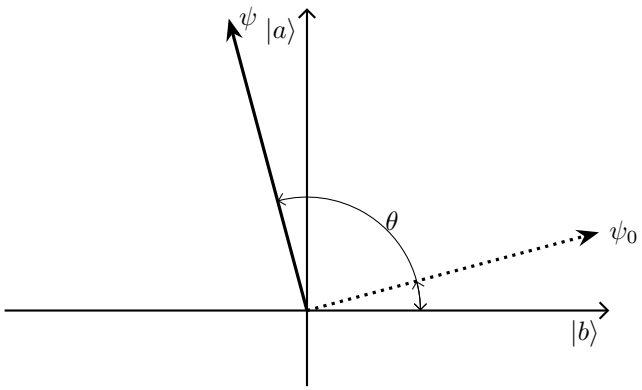
So, what may be happening has been attempted to be explained with the following diagrams.
After $\sqrt{N}$ iterations, the state ψ is quite close to the desirable state. The probability of collapsing to the desirable state is as high as $80\% - 90\%$ (usual number as seen from the simulator code).



Now, when we do amplitude negation by reflecting about $|b\rangle$, we get state as shown.



And when we do the inversion about mean, we get the state as shown.



So exceeding one query does not make much difference, but when we do this a few number of times, ψ keeps on deviating from the required state. And then after some number of such iterations, it again starts to return to the position. And this "some number of iterations" is less than $\sqrt{N}$ because since the angle made from $|b\rangle$ is large, the iterations of the algorithm make huge changes in the states and thus requires fewer steps.

Note - This was observed when there was only one desirable state. With multiple desirable states a similar pattern was followed but not accurately.

5.2 Assumption about the given black box

It is assumed that a quantum circuit has already been implemented that identifies the answer that we use as a black box. Well, if such a black box has been implemented, the maker of the black box would already know about the desired state.

I think that maybe in future a situation may arise when the implemented black box is provided by nature and the algorithm may be used to find the desirable state.

But nevertheless, this amazing algorithm does given an idea about the power of Quantum Computing.