Grover's Algorithm

Paolo Bettelini

Contents

1 Introduction

Given a list of N element, an item ω with a unique properties, on average we will need to check $\frac{N}{2}$ elements before finding ω. This classical computation is $O(n)$ in time complexity.

Grover's algorithm reduces this time complexity to $O(\sqrt{n})$, meaning that if we have a list of size 100 it will take 10 steps to find ω instead of 50 on average.

This quantum algorithm uses amplitude amplification of a superposition to have a near perfect probability of finding ω.

2 Algorithm

The list of elements is comprised of all the possible computational basis states the qubits can be in. For example ($|0\rangle \;\Rightarrow |255\rangle$) for 8 qubits.

The oracle U_ω negates the phase of the state if it is not ω.

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

We define a function $f(x)$ such that the output is 1 if ($x = \omega$), 0 otherwise.

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

The oracle applied to a state is given by $U_\omega|x\rangle = (-1)^{f(x)}|x\rangle$

The oracle can be represented with a diagonal matrix, where only the position of ω has a negative phase. The diagonal is made of 1s except for the entry of ω, -1.

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