Grover's Algorithm

Goal:

- To search a large, unsorted database more efficiently

Uses:

- Search an unsorted database
- Finding a key to decrypt an encrypted message
- Mean and median estimation
- Solving the collision problem
- Solving NP-complete problems

Basics:

Database can be thought of as a function(C(x))
from a set X to a set Y,
where X and Y both have a linear order

Ex: either (X_1 <= X_2) or (X_2 <= X_1)

Grover's algorithm tries to reduce the amount of, C, or steps to find an element.

Example:

If X is a set of 100,000 phone numbers and Y is the corresponding set of names:

To find Bob in Y, using a classical computer would take an average of 50,000 steps.

To find Bob using Grover's would take an average of 100 steps.

The Algorithm:

Assumptions:

You have a system with $N = 2^n$ states
There is a unique S_m that satisfies $C(S_m) = 1$
For all other states $C(S) = 0$
C can be calculated in unit time

Step 1: Use the Walsh-Hadamard operator to set all the states in the register to an equal superposition.

Step 2: Repeat the following $(n^{1/2})$ times:

- If $C(S) = 1$, rotate the phase π radians, else leave untouched
- Apply the inversion about average operator to the register

Step 3: Collapse the state of the register to get the n bit label of the S_m state with a certain probability.

4 Bit Example:

Given a quantum register of 4 qubits in the state (1, 0, 0, 0), find the third element.

Step 1: Perform the Walsh-Hadamard transformation

State = (.5, .5, .5, .5)

Step 2_1: Rotate the marked element

State = (.5, .5, -.5, .5)

Step 2_2: Apply the inversion about the average

State = (0, 0, 1, 0)

Step 3: Measure the state

Measured state 3 with probability of 1