Review
000000000

Introduction
0000000000000000000000000000000

Programming Grover's Algorithm
000000

Grover's Search Algorithm

Julie Butler

Review
000000000

Introduction
00000000000000000000000000000

Programming Grover's Algorithm
000000

Outline

- ▶ Day 1
  - ▶ Review
    - ▶ Math and Quantum Mechanics Review
  - ▶ Introduction to the Algorithm
    - ▶ Classical Search Algorithms
    - ▶ Visually Implement Grover's
- ▶ Day 2
  - ▶ Coding the Two Qubit Case
  - ▶ Coding the General Case (Three Ways)

◀ □ ▶ ◀ 🗗 ▶ ◀ 🗉 ▶ ◀ 🗉 ▶   🗉   ꘯ꝗꝗ

Review
●○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

Review

Identity Matrix

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

▶ The identity matrix shown above is for 4x4 matrices, general pattern extends to matrices of all size
   ▶ 1's on main diagonal, 0's everywhere else

Grover's Search Algorithm

Review
000●00000

Introduction
00000000000000000000000000000

Programming Grover's Algorithm
000000

States are Vectors, Gates are Matrices

▶ Quantum mechanical states are represented by vectors, describe the entire system
  ▶ One qubit states are length 2, two qubit states are length 2, n qubit states are 2^n
▶ Quantum computing gates are represented by **unitary** matrices
  ▶ One qubit gates are 2x2, two qubit gates are 4x4, n qubit gates are $2^n x 2^n$
▶ Note that many multi-qubit gates involve one or more control qubits and one or more target qubits:
  ▶ Fredkin Gate (one control, two targets)
  ▶ Three-Qubit X Gate (two controls, one target)

Review
000●0000

Introduction
0000000000000000000000000000000

Programming Grover's Algorithm
000000

Statevectors Are Made of Basis States

▶ Consider the following state vector:
$|\psi\rangle = \frac{1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$

▶ The amplitude of the $|00\rangle$ state is $\frac{1}{2}$

▶ The probability of measuring $|\psi\rangle$ and obtaining $|00\rangle$ is $|\frac{1}{2}|^2 = \frac{1}{4}$

▶ Consider the following state vector:
$|\psi\rangle = \frac{-1}{2}|00\rangle + \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle + \frac{1}{2}|11\rangle$

▶ The amplitude of the $|00\rangle$ state is $\frac{-1}{2}$

▶ The probability of measuring $|\psi\rangle$ and obtaining $|00\rangle$ is $|\frac{-1}{2}|^2 = \frac{1}{4}$

Review
○○○○●○○○○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

Define Multiple Gate Result Through Matrix Multiplication
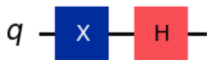
Figure 1: One Qubit Circuit with Two Gates

▶ Expected Output?
▶ Now show it one gate at a time.
▶ Now show it with a combination gate

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Grover's Search Algorithm

Review
○○○○○●○○○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

Overlap of Wavefunctions

▶ For an inner product where one of the states is a statevector and the other state is a basis state, the calculation is called an overlap
  ▶ It determines the portion of the statevector in the direction of the basis vector

▶ Example: Consider the following basis in Cartesian coordinates

$$|x\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |y\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad |\phi\rangle = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

▶ Then $\langle y|\phi\rangle = 4$ which is the overlap between $|y\rangle$ and $|\phi\rangle$

Review
○○○○○○●○○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

Outer Product

▶ An outer product takes two vectors and creates a matrix (an inner product takes two vectors and creates a scalar).

▶ Consider the following two vectors of length 3:

$$|x\rangle = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} \quad |y\rangle = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \end{bmatrix}$$

▶ The outer product between these two vectors, represented as $|x\rangle\langle y|$ is

$$|x\rangle\langle y| = \begin{bmatrix} x_0 y_0^* & x_0 y_1^* & x_0 y_2^* \\ x_1 y_0^* & x_1 y_1^* & x_1 y_2^* \\ x_2 y_0^* & x_2 y_1^* & x_2 y_2^* \end{bmatrix}$$

Grover's Search Algorithm

Review
○○○○○○○●○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

Completeness Relation

▶ If the states $|x_i\rangle$ form a complete basis then:

$$\sum_i |x_i\rangle\langle x_i| = \mathbb{1}$$

Review
○○○○○○○○●

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

Projection Operator

▶ A useful application of the outer product is in the construction of projection operators.

▶ The projection operator *projects* out a specific component of a of a state related to the basis state that was used to the projection operator.

▶ Example: Consider the following basis in Cartesian coordinates

$$|x\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |y\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad |\phi\rangle = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

▶ $|y\rangle\langle y|$ is the projection operator for $|y\rangle$, so

$$|y\rangle\langle y|\phi\rangle = |y\rangle 4 = 4|y\rangle = \begin{bmatrix} 0 \\ 4 \end{bmatrix}$$

Grover's Search Algorithm

Review
000000000

Introduction
●0000000000000000000000000000

Programming Grover's Algorithm
000000

Introduction

Review
○○○○○○○○○

Introduction
○●○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

Classical Linear Search

▶ If the data is sorted there are a variety of algorithms which have good performance
▶ If the data is unsorted you are left essentially with brute force
  ▶ Average performance is $\frac{N}{2}$ searches
  ▶ Worst case is N searches
▶ Computational Performance: $O(N)$

Review
○○○○○○○○○

Introduction
○○●○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

A New Way of Thinking of the Algorithm

▶ Standard algorithm returns the index of the wanted element
▶ Consider instead marking the indices corresponding to the wanted element

```python
def linear_search_version_2 (my_list, element):
    length = len(my_list)
    results = np.zeros(length)
    for i in range(length):
        if my_list[i] == element:
            results[i] = 1
    return results
```

Review
000000000

Introduction
0000●0000000000000000000000000

Programming Grover's Algorithm
000000
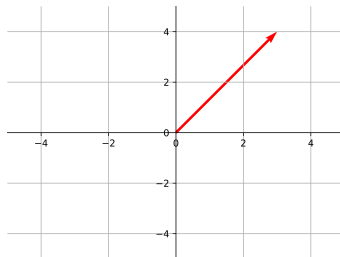
Reflections

Figure 2: Reflection About the X Axis

Review
000000000

Introduction
0000●00000000000000000000000000

Programming Grover's Algorithm
000000
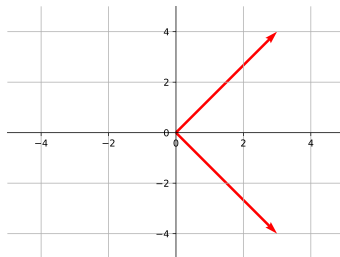
Reflections

Figure 3: Reflection About the X Axis

Review
000000000

Introduction
0000000●000000000000000000000000

Programming Grover's Algorithm
000000
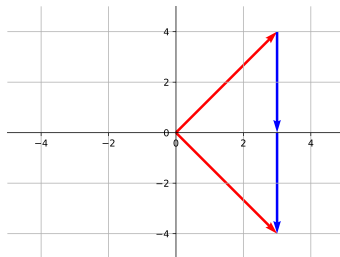
Reflections

Figure 4: Reflection About the X Axis

Review
000000000

Introduction
00000●0000000000000000000000

Programming Grover's Algorithm
000000

Mathematical Representations of Reflections

$$\begin{bmatrix} 3 \\ 4 \end{bmatrix} - \begin{bmatrix} 0 \\ 4 \end{bmatrix} - \begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \\ -4 \end{bmatrix}$$

$$\begin{bmatrix} 3 \\ 4 \end{bmatrix} - 2 \begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} 3 \\ -4 \end{bmatrix}$$

▶ But consider the following representation:

$$|x\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad |y\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad |\phi\rangle = 3|x\rangle + 4|y\rangle$$

Mathematical Representations of Reflections

▶ Then we can represent the reflection as:

$$ref_x(|\phi\rangle) = |\phi\rangle - 4|y\rangle - 4|y\rangle = |\phi\rangle - 2(4|y\rangle)$$

▶ But remember that from our discussion of overlaps, $\langle y|\phi\rangle = 4$, so

$$ref_x(|\phi\rangle) = |\phi\rangle - 2|y\rangle\langle y|\phi\rangle = (\mathbf{I} - 2|y\rangle\langle y|)|\phi\rangle$$

▶ Where we are reflecting around the x-axis, so y is the other axis

Review
000000000

Introduction
0000000●0000000000000000000

Programming Grover's Algorithm
000000

Mathematical Representations of Reflections

▶ For a reflection in more than two-dimensional space we would need to subtract the outer products of all other basis vectors besides the axis we are rotating around:

$$ref_x(|\phi\rangle) = (\mathbf{I} - 2|y\rangle\langle y| - 2|z\rangle\langle z|)|\phi\rangle$$

▶ This can be tedious for many basis vectors, but remember that the identity matrix can be represented as a sum of outer products of all basis vectors. So in 3D Cartesian:

$$ref_x(|\phi\rangle) = (\mathbf{I} - 2|y\rangle\langle y| - 2|z\rangle\langle z|)|\phi\rangle$$
$$= (|x\rangle\langle x| + |y\rangle\langle y| + |z\rangle\langle z| - 2|y\rangle\langle y| - 2|z\rangle\langle z|)|\phi\rangle$$
$$= (|x\rangle\langle x| - |y\rangle\langle y| - |z\rangle\langle z|)|\phi\rangle$$

Review
000000000

Introduction
0000000000●00000000000000000000

Programming Grover's Algorithm
000000

Mathematical Representations of Reflections

▶ Now add and subtract by the outer product of the axis we are rotating around:

$$ref_x = (|x\rangle\langle x| + |x\rangle\langle x| - |x\rangle\langle x| - |y\rangle\langle y| - |z\rangle\langle z|)|\phi\rangle$$
$$= (2|x\rangle\langle x| - (|x\rangle\langle x| + |y\rangle\langle y| + |z\rangle\langle z|)|\phi\rangle$$
$$= (2|x\rangle\langle x| - \mathbf{I})|\phi\rangle$$

▶ Generalization: To rotate a statevector $|\phi\rangle$ around any axis k:

$$ref_k(|\phi\rangle) = (2|k\rangle\langle k| - \mathbf{I})|\phi\rangle$$

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○●○○○○○○○○○○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

Parts of Grover's Algorithm

Figure 5: Grover's Algorithm

Review
000000000

Introduction
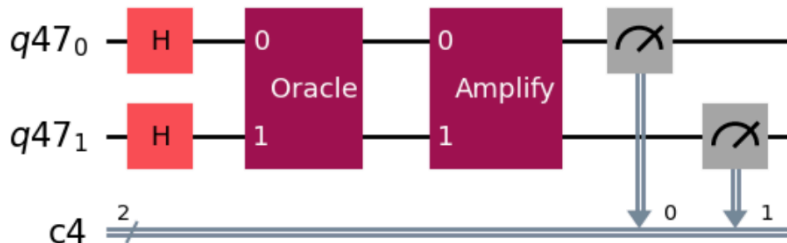00000000000●00000000000000000

Programming Grover's Algorithm
000000

Hadamard Gates

▶ First step is to apply a Hadamard gate to every qubit, creating a superposition of every possible state

$$H^{\otimes n}|0^n\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle = |\phi\rangle$$

Grover's Search Algorithm

Review
000000000

Introduction
000000000000●0000000000000000

Programming Grover's Algorithm
000000

Oracle

▶ The goal of the algorithm is to flip the sign on the amplitude of the desired state(s)
  ▶ Grover's can be used to find more than one desired solution at a time
▶ The oracle should flip the sign of the amplitude for all desired states but leave all other amplitudes unchanged

Review
000000000

Introduction
0000000000000●000000000000000

Programming Grover's Algorithm
000000

Amplification Algorithm (Diffusion Algorithm)

▶ A negative amplitude gives the same probability as a positive amplitude, so we need another step in the algorithm

▶ The amplification algorithm (amplitude amplification algorithm or diffusion algorithm) increase the amplitude on the marked states but decreases the amplitude on the unmarked states

▶ **NOTE** the oracle and amplification algorithm may have to be applied more than once to achieve significant separation in amplitudes between the marked and unmarked states.

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○●○○○○○○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

$O(N)$ vs. $O(\sqrt{N})$ Run Times
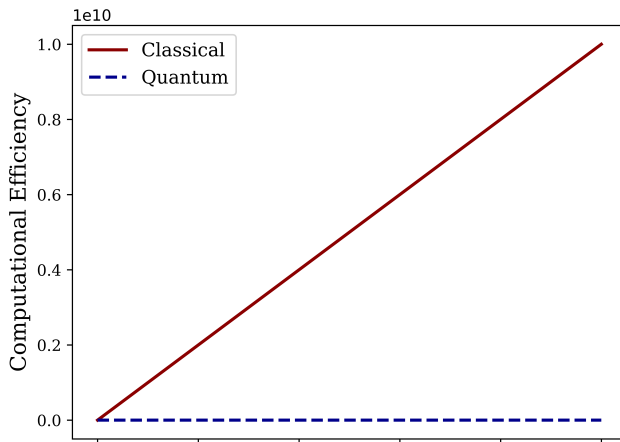
Visual Walkthrough

▶ In the next few slides we will visually walkthrough what Grover's algorithm is doing as it processes a system.

▶ The basic steps are to negate the amplitude of the desired state (oracle) and then reflect the entire state around the mean amplitude (amplification)
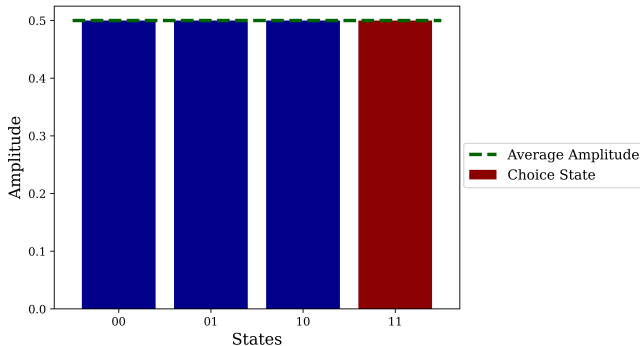
Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○●○○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

Step 1: Starting Point for Two Qubits

Figure 7: Grover's Algorithm Visualization Step 1

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○●○○○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

Step 2: Negate Amplitude of Desired State
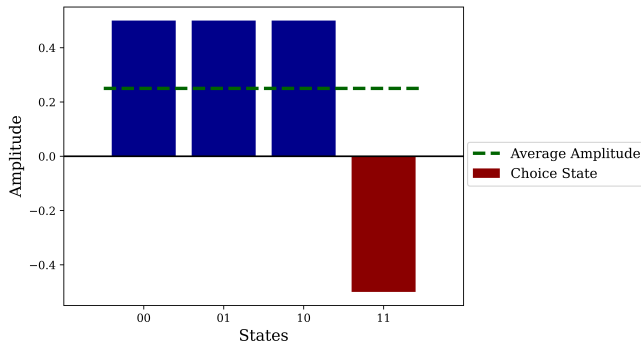
Figure 8: Grover's Algorithm Visualization Step 2

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○●○○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

Step 3: Reflect Around Mean Amplitude

Figure 9: Grover's Algorithm Visualization Step 3
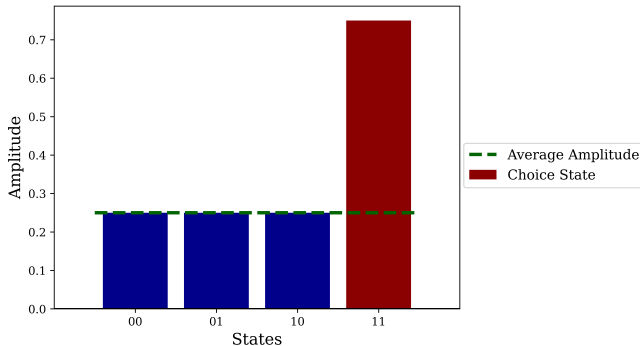
Review
ooooooooo

Introduction
ooooooooooooooooooo●oooooooooo

Programming Grover's Algorithm
oooooo

Step 4: Negate Amplitude of Desired State
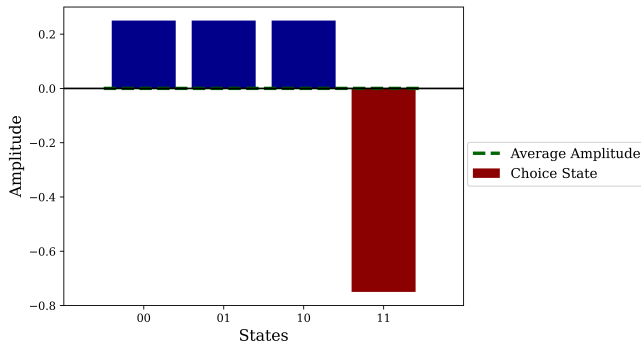
Figure 10: Grover's Algorithm Visualization Step 4

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○○○○●○○○○○○○○○○

Programming Grover's Algorithm
○○○○○○

Step 5: Reflect Around Mean Amplitude
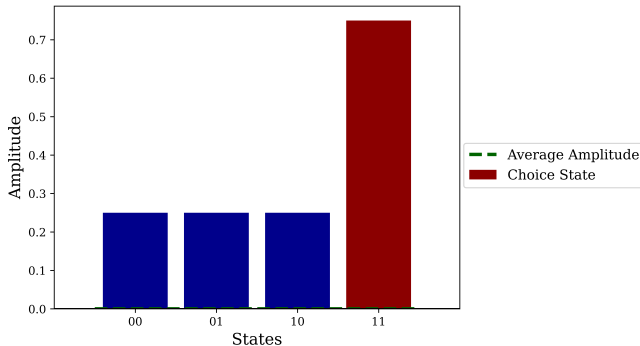
Figure 11: Grover's Algorithm Visualization Step 5

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○○○○●○○○○○○○○

Programming Grover's Algorithm
○○○○○○

How Many Times To Run $\longrightarrow$ Grover's Circle

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○●○○○○○○○

Programming Grover's Algorithm
○○○○○○
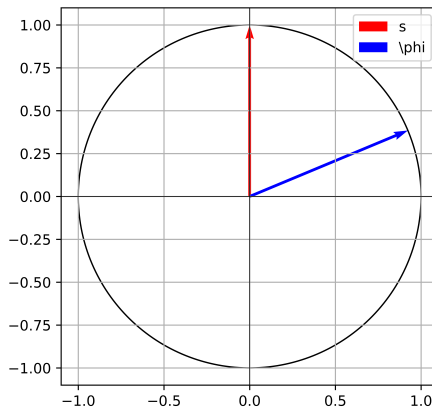
Add $|s^\perp\rangle$, a state which is perpendicular to $|s\rangle$
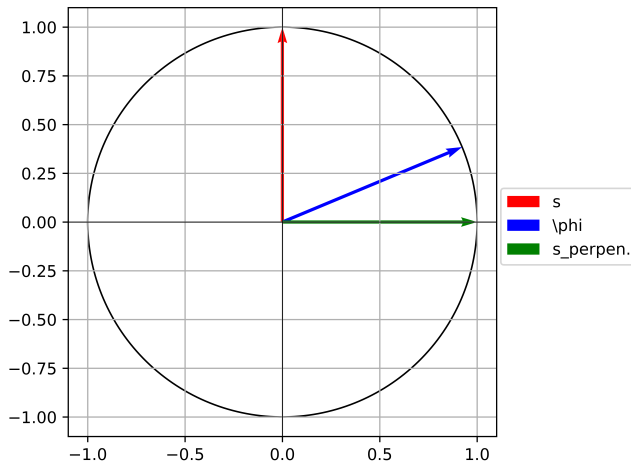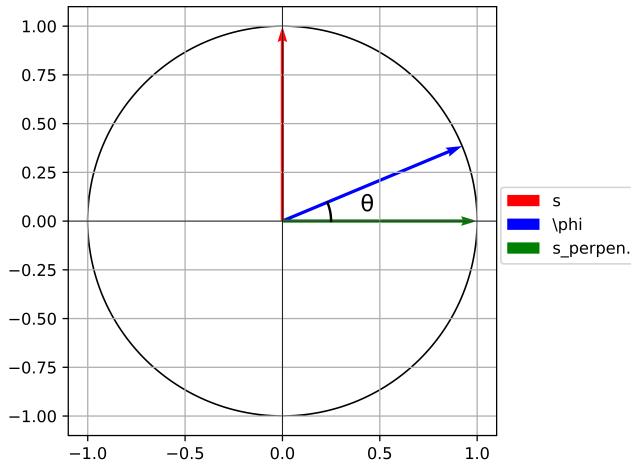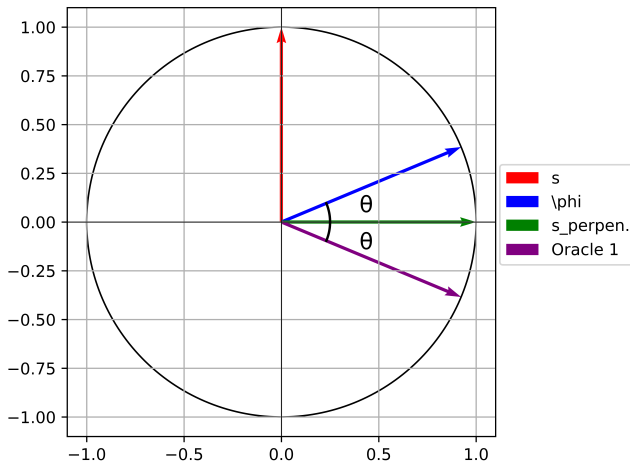
Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○●○○○○○○○

Programming Grover's Algorithm
○○○○○○

Note the angle θ

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○●○○○○○

Programming Grover's Algorithm
○○○○○○

Negate the component of $|\phi\rangle$ that comes from $|s\rangle$ (Oracle)

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○○○●○○○○

Programming Grover's Algorithm
○○○○○○

Reflect around the original value of $|\phi\rangle$ (Amplification)

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○○○○●○○○

Programming Grover's Algorithm
○○○○○○

Negate the component of $|\phi\rangle$ that comes from $|s\rangle$ (Oracle)

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○○○●○○

Programming Grover's Algorithm
○○○○○○

Reflect around the original value of $|\phi\rangle$ (Amplification)

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○●○

Programming Grover's Algorithm
○○○○○○
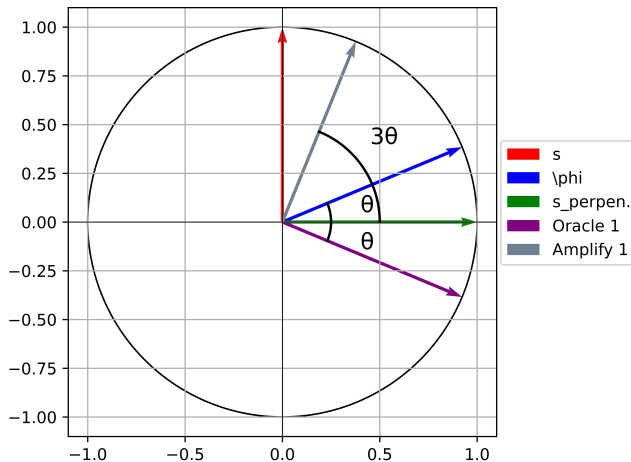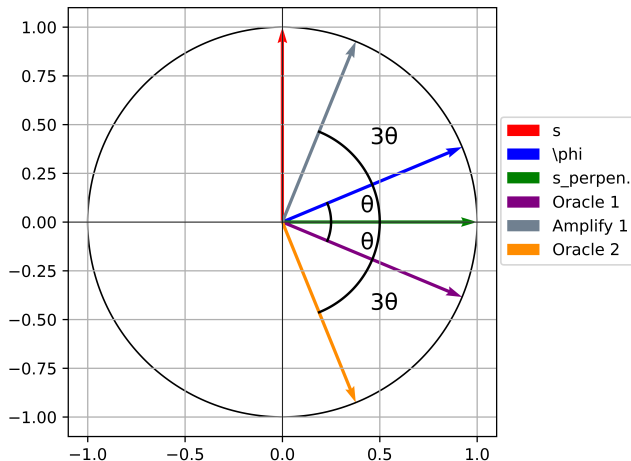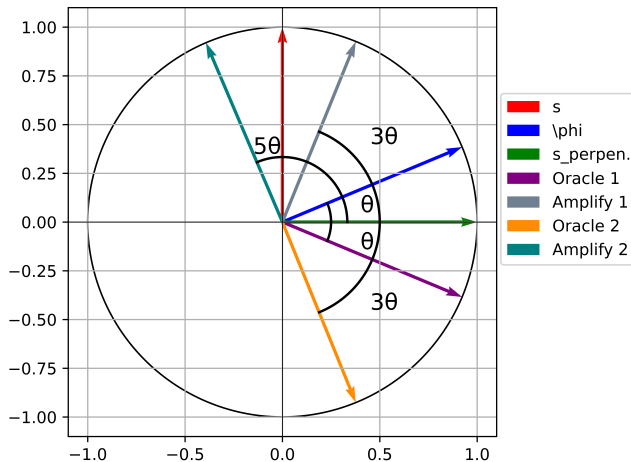
How Many Times To Run $\longrightarrow$ Grover's Circle

▶ Now some trigonometry

$$sin\theta = \frac{1/\sqrt{N}}{1} \longrightarrow \langle s|\phi \rangle = \frac{1}{\sqrt{N}}, \ |\langle \phi|\phi \rangle|^2 = 1$$

$$\theta = arcsin(\frac{1}{\sqrt{N}}) \approx \frac{1}{\sqrt{N}}$$

$$(2t+1)\theta = \frac{\pi}{2} \longrightarrow t = \frac{\pi}{4}\sqrt{N} - \frac{1}{2}$$

Grover's Search Algorithm

Review
000000000

Introduction
000000000000000000000000000000●

Programming Grover's Algorithm
000000

How Many Times To Run $\longrightarrow$ Grover's Circle

▶ In general the algorithm will run $\frac{\pi}{4}\sqrt{\frac{N}{M}}$ times where there are N items in the search space and the algorithm is looking for M different items.
  ▶ Typically this equation is "floored" to avoid overshooting

Review
○○○○○○○○○

Introduction
○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○

Programming Grover's Algorithm
●○○○○○

Programming Grover's Algorithm

Two Qubit Case

▶ Let's consider a two-qubit Grover's algorithm
  ▶ The "good state" is $|11\rangle$
  ▶ The other states are the "bad states"

◀ □ ▶ ◀ 𝕗 ▶ ◀ ≡ ▶ ◀ ≡ ▶   ≡   ⟳ Q ⟳

Review
000000000

Introduction
0000000000000000000000000000000

Programming Grover's Algorithm
000●000

What Could the Oracle Be?

▶ General form: $U_s = \mathbf{I} - |s\rangle\langle s|$
▶ What gate or combination of a gates would turn $|11\rangle \longrightarrow -|11\rangle$ but leave all other states unchanged?
   ▶ Controlled Z!

Grover's Search Algorithm

Review
000000000

Introduction
0000000000000000000000000000

Programming Grover's Algorithm
000●00

What Could the Amplification Be?

$$U_d = 2|\phi\rangle\langle\phi| - \mathsf{I}$$

▶ Remember this step is a rotation around the mean amplitude
▶ But since $|\phi\rangle$ is a superposition of all possible states:

$$H^{\otimes n}(2|0\rangle\langle0| - \mathsf{I})H^{\otimes n}$$

▶ Very similar implementation as the Oracle, same gates will be involved

Grover's Search Algorithm

Review
000000000

Introduction
0000000000000000000000000000000

Programming Grover's Algorithm
0000●0

Amplification Algorithm

1. Hadamard to all qubits
2. NOT to all qubits
3. Multi-controlled-Z where all top qubits are the control and the bottom qubit is the target
4. NOT to all qubits
5. Hadamard to all qubits

General Case

▶ The amplification algorithm will remain the same, but what could be a general form for the oracle?

▶ It must have a multi-controlled-Z gate, only other change is to add leading and post NOT gates if the digit is a zero

1. For any digit in the marked string that is a 0, apply a NOT gate to the corresponding qubit.
2. Multi-controlled-Z where all top qubits are the control and the bottom qubit is the target
3. For any digit in the marked string that is a 0, apply a NOT gate to the corresponding qubit.

▶ **To Qiskit!**