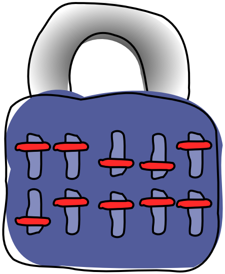
CPEN 400Q Lecture 09
Grover's algorithm

Wednesday 5 February 2025

Announcements

- Midterm grades available tomorrow; can pick up at my office 2pm or later (or Friday, and will bring to class on Monday)
- Project details next week (create 7 groups of 4, 2 groups of 3)
- First literacy assignment and A2 available soon
- Quiz 4 beginning of class Monday (about *this week*'s material)

We modeled breaking a (binary) lock as a function:

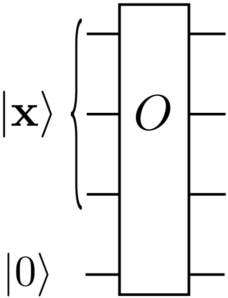


Trying combination $\mathbf{x}$ modeled as querying an *oracle* that evaluates $f(\mathbf{x})$. The number of queries is an algorithm's *query complexity*.

Image credit: Codebook node A.1

Last time

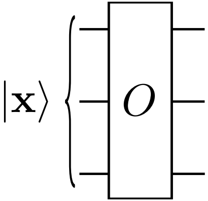
We expressed oracle queries as quantum circuits in two ways.



$$
\begin{aligned}
O|000\rangle|0\rangle &= |000\rangle|0\rangle \\
O|001\rangle|0\rangle &= |001\rangle|0\rangle \\
O|010\rangle|0\rangle &= |010\rangle|0\rangle \\
O|011\rangle|0\rangle &= |011\rangle|0\rangle \\
O|100\rangle|0\rangle &= |100\rangle|0\rangle \\
O|101\rangle|0\rangle &= |101\rangle|0\rangle \\
O|110\rangle|0\rangle &= |110\rangle|1\rangle \\
O|111\rangle|0\rangle &= |111\rangle|0\rangle
\end{aligned}
$$

Last time

We expressed oracle queries as quantum circuits in two ways.

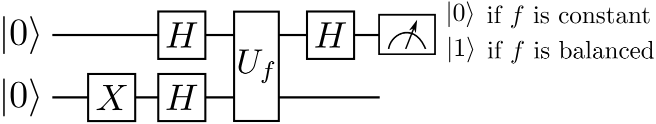


$$
\begin{aligned}
O|000\rangle &= |000\rangle\\
O|001\rangle &= |001\rangle\\
O|010\rangle &= |010\rangle\\
O|011\rangle &= |011\rangle\\
O|100\rangle &= |100\rangle\\
O|101\rangle &= |101\rangle\\
O|110\rangle &= -|110\rangle\\
O|111\rangle &= |111\rangle
\end{aligned}
$$

Last time

We applied Deutsch's quantum algorithm to determine if a function is *constant* or *balanced* using one oracle query

Name	Action	Name	Action
f_1	$f_1(0) = 0$ $f_1(1) = 0$	f_2	$f_2(0) = 1$ $f_2(1) = 1$
f_3	$f_3(0) = 0$ $f_3(1) = 1$	f_4	$f_4(0) = 1$ $f_4(1) = 0$

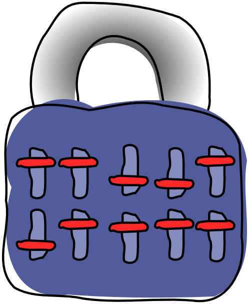


This is a quantum speedup; classical case requires 2 queries.

Learning outcomes

- Describe the strategy of amplitude amplification
- Visualize Grover's algorithm in two different ways
- Implement basic oracle circuits in PennyLane
- Implement Grover's search algorithm

Let's break that lock!



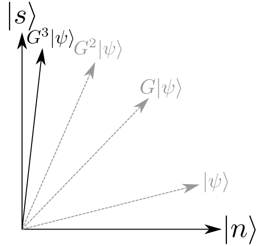
Classical: in the worst case, oracle queries

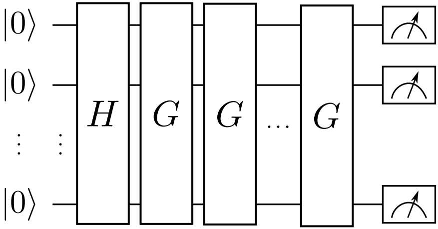
Quantum: queries with Grover's algorithm

Image credit: Codebook node A.1

Grover's quantum search algorithm

Grover's search algorithm starts with a uniform superposition, then *amplifies* the amplitude of the state corresponding to the solution.





In other words, we want to go from the uniform superposition

to something like this:

Assume we have an oracle that performs

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

Start with the uniform superposition.

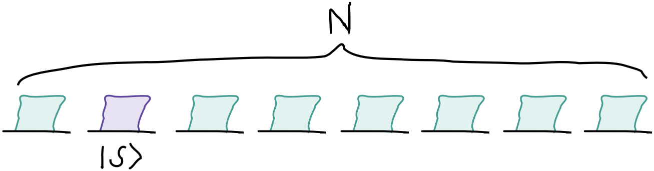


Image credit: Codebook node G.1

Applying the oracle flips the sign for the solution state:

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

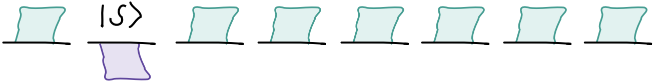
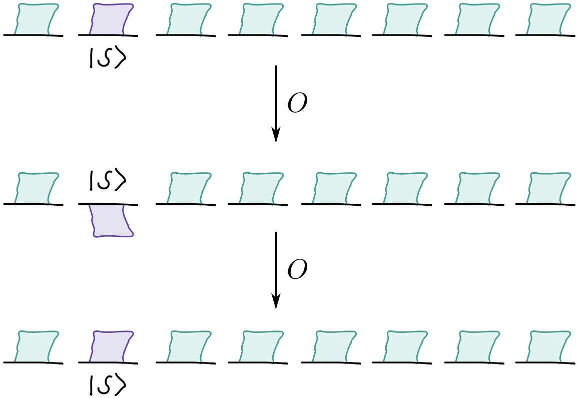


Image credit: Codebook node G.1

Now what?

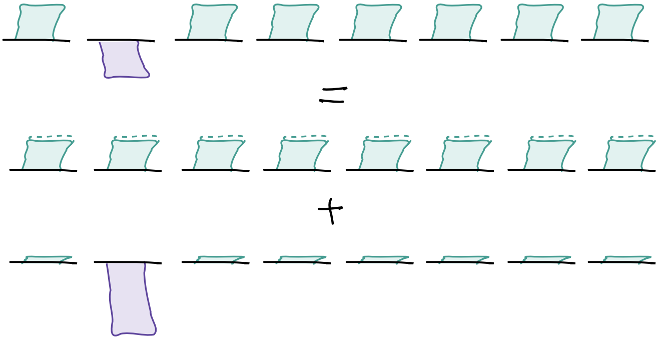


Can't just apply the oracle again... need to do something different.

Image credit: Codebook node G.1

Grover's algorithm: amplitude visualization

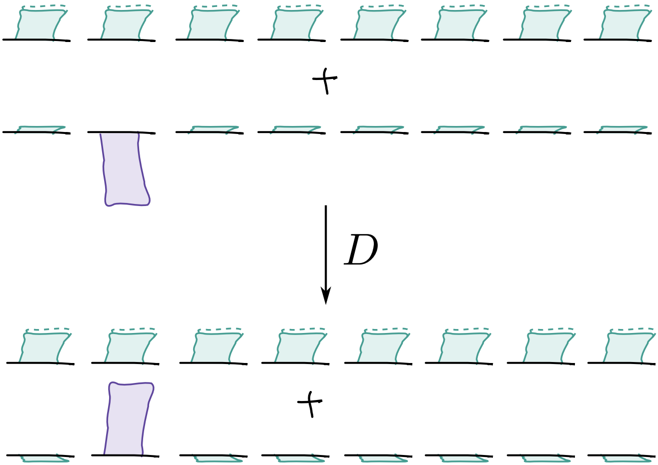
Let's write the amplitudes in a different way:



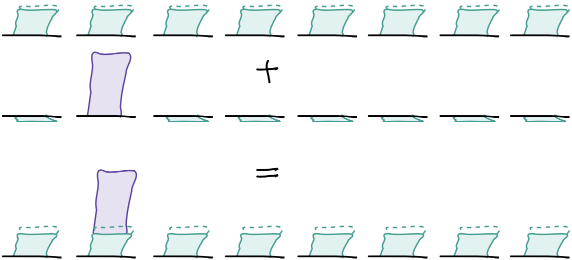
Why does this help?

Image credit: Codebook node G.1

What if we had an operation that would flip everything in the second part of the linear combination?



Let's add these back together...



We have "stolen" some amplitude from the other states, and added it to the solution state!

Image credit: Codebook node G.1

Grover's algorithm: amplitude visualization

Doing this sequence once is one "iteration":

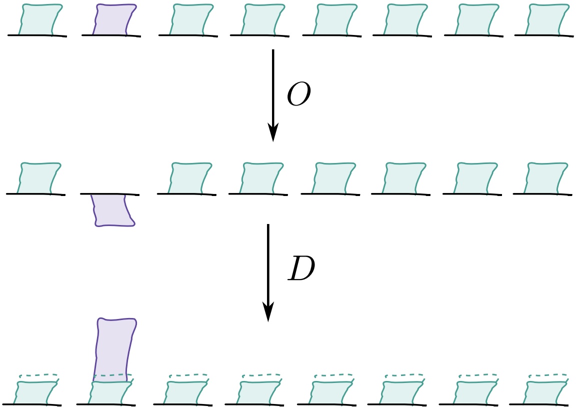
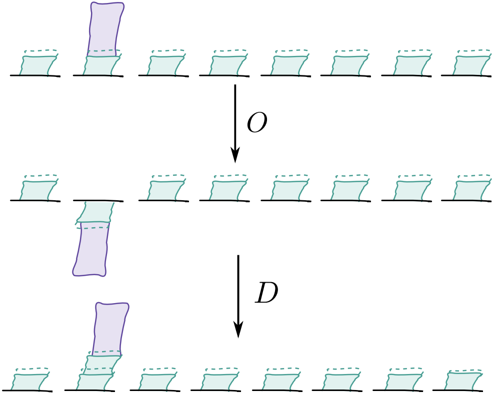


Image credit: Codebook node G.1

Grover's algorithm: amplitude visualization

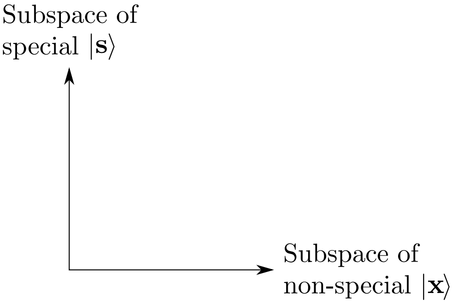
If we do it again, we can steal even more amplitude!



Grover's algorithm works by applying O then D multiple times, until the probability of observing the solution state is maximized.

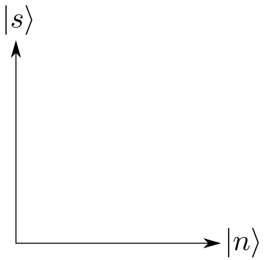
Image credit: Codebook node G.1

Grover's algorithm: geometric visualization



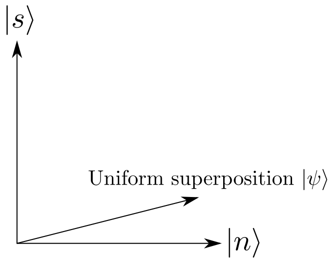
Partition the computational basis states into two subspaces:

1. The special state $|\mathbf{s}\rangle$
2. All the other states



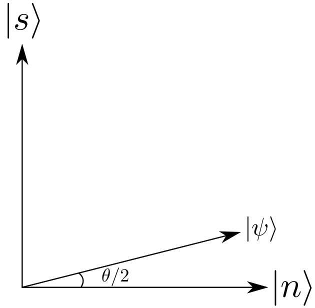
Let's write these out as superpositions:

Grover's algorithm: geometric visualization



$$
\begin{aligned}
|s\rangle &= |\mathbf{s}\rangle \\
|n\rangle &= \frac{1}{\sqrt{2^n - 1}} \sum_{\mathbf{x} \neq \mathbf{s}} |\mathbf{x}\rangle
\end{aligned}
$$

We can write the uniform superposition in terms of $|s\rangle$ and $|n\rangle$:

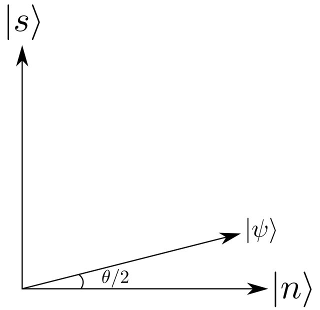


Instead of working with these complicated coeffients:

$$|\psi\rangle = \frac{1}{\sqrt{2^n}}|s\rangle + \frac{\sqrt{2^n - 1}}{\sqrt{2^n}}|n\rangle,$$

rexpress in terms of an angle θ:

Grover's algorithm: geometric visualization



Want to apply operations to

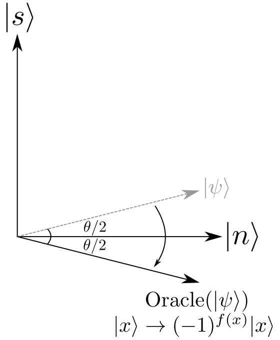
$$|\psi\rangle = \sin\left(\frac{\theta}{2}\right)|s\rangle + \cos\left(\frac{\theta}{2}\right)|n\rangle$$

to increase the amplitude of $|s\rangle$ while decreasing that of $|n\rangle$.

Two steps:

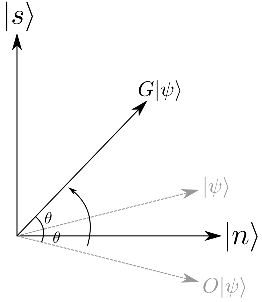
1. Apply the oracle O to 'pick out' the solution
2. Apply a 'diffusion operator' D to adjust the amplitudes.

Grover's algorithm: geometric visualization



The oracle *flips* the amplitude of the special basis states in $|\psi\rangle$.

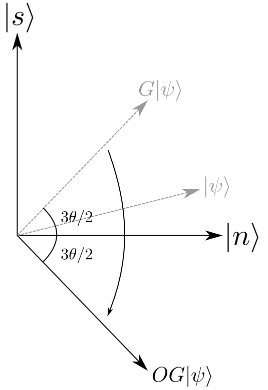
We can visualize this as a *reflection about the subspace* of non-special elements.



The diffusion operator is a bit less intuitive - it performs a *reflection about the uniform superposition* state.

A full Grover iteration $G = DO$ sends

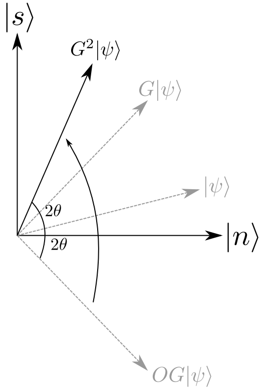
$$G\left(\sin\left(\frac{\theta}{2}\right)|s\rangle + \cos\left(\frac{\theta}{2}\right)|n\rangle\right) =$$



Now we repeat this...

Apply the oracle and reflect about the non-special elements.

Grover's algorithm: geometric visualization

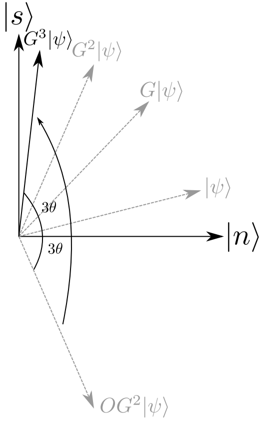


Apply the diffusion operator and reflect about the uniform superposition to boost the amplitude of the special state.

Grover's algorithm: geometric visualization

After k Grover iterations we will have the state

$$G^k|\psi\rangle =$$



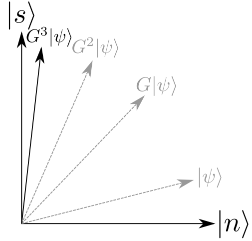
It *is* possible to over-rotate! We can differentiate to find the optimal k:

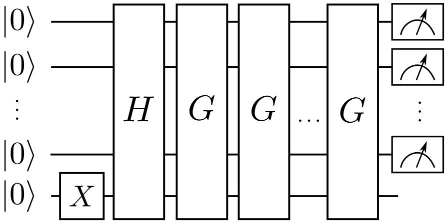
$$k \leq \left\lceil \frac{\pi}{4}\sqrt{2^n} \right\rceil$$

After k operations we will be most likely to obtain the special state when we measure.

Multiple approaches depending on the format of the oracle. We will use this one:

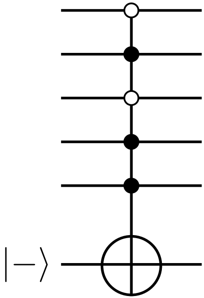
$$O|\mathbf{x}\rangle|y\rangle = |\mathbf{x}\rangle|y \oplus f(\mathbf{x})\rangle$$



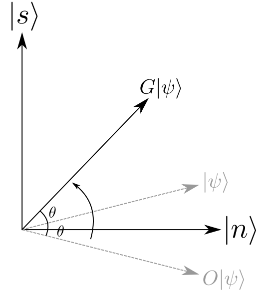


What do circuits for the oracle and diffusion look like?

Exercise: show that a multicontrolled X gate, controlled on **s**, can be used as an oracle:



The diffusion operator performs a reflection about the uniform superposition state.

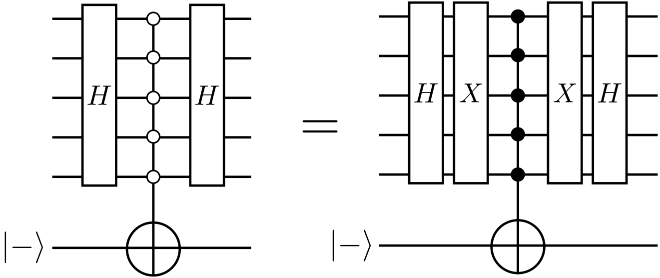


Exercise: Show that the unitary matrix given by

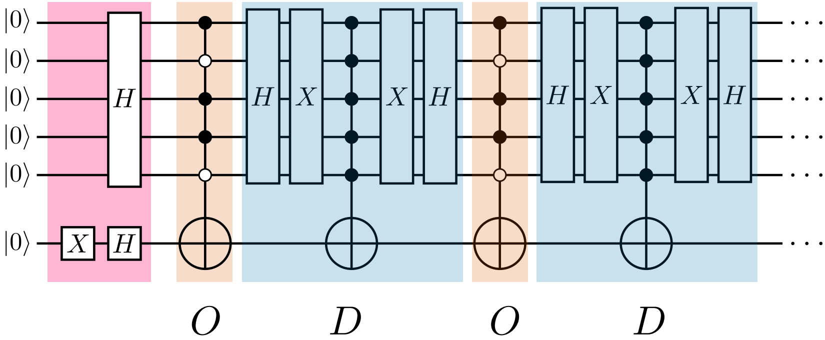
$$D = 2|\psi\rangle\langle\psi| - I$$

is equivalent to the diffusion operator.

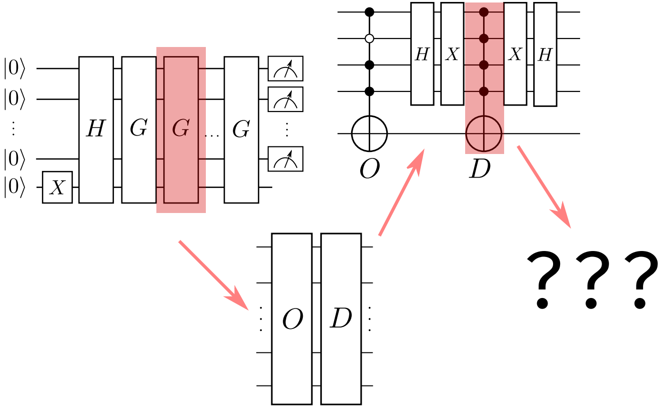
The diffusion circuit



The full Grover circuit



Clearly, each of the $O(\sqrt{2^n})$ queries requires some number of gates... how much does Grover *really* cost?



Next class: look inside the black box!

Next time

Content:

- Introduction to quantum compilation and resource estimation
- Quiz 4

Action items:

1. Literacy assignment 1 (when available)
2. Assignment 2 (when available)

Recommended reading:

- For this class: Codebook module GA; Nielsen & Chuang 6.1
- For next class: Nielsen & Chuang 4.5