Notes 8.370/18.435 Fall 2022

Lecture 24 Prof. Peter Shor

Today, we are talking about Grover's search algorithm. This will let you search a possible space of size N for a solution in $\sqrt{N}$ time.

Suppose you have an equation, and you want to know whether it has a solution in some finite search space of possible solutions. To make this more concrete, suppose you were looking at the equation $x^3 + y^3 + z^3 = 42$, where x, y, and z are integers. Number theorists have studied this type of equation, and equations like this often have small solutions when you replace 42 with a different number, like 34:

$$3^3 + 2^3 + (-1)^3 = 34$$
$$5^3 + (-4)^3 + (-3)^3 = 34.$$

However, nobody had discovered three cubes that added to 42 until a few years ago, when Andrew Booker (a math professor at U. Bristol) and Drew Sutherland (a math professor at MIT) used a large amount of computer time to find three 17-digit numbers whose cubes added up to 42.

How would you solve this problem with brute force search? You would take two numbers, x, and y, and see whether x^3+y^3 had an integer cube root. This would require searching around 10^{34} potential solutions before you found the right 17-digit numbers. Using Grover's algorithm would only require around the square root of that, so the time it would take to check around 10^{17} numbers on a classical computer. Booker and Sutherland, of course, didn't search 10^{34} potential solutions ... they used an algorithm that was much cleverer than that, and they still needed a million hours of computer time (run on computers that would otherwise have been idle). But a quantum computer could have reduced the time take substantially without any cleverness at all. Furthermore, quite often you can combine clever classical techniques with ideas from Grover's algorithm to reduce the computational time still further than plain Grover search.

So how does Grover's algorithm work? We are given a phase oracle that will tell us whether something is a solution:

$$O_p \,|\, x\rangle = \begin{cases} -\,|\, x\rangle & \text{if x is a solution,} \\ \;\;\,|\, x\rangle & \text{otherwise.} \end{cases}$$

If you have a circuit that checks whether something is a solution, it is fairly straightforward to implement this oracle. You choose a qubit that you set to $|\, 0\rangle$ if the input is not a solution and $|\, 1\rangle$ if the input is a solution, you then apply σ_z to this qubit, and finally you uncompute everything so you are left only with the input $\pm\,|\, x\rangle$.

We will encode the elements of the search space by the numbers 0, 1, 2, 3, ..., $N-1$. We can assume that N is a power of 2 by adding dummy elements of the search space ... the running time of the algorithm isn't greatly affected by rounding N up to a power of 2. Grover's algorithm works by starting in the superposition of every item in the search space:

$$|\, \psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^{N-1} |\, i\rangle \,.$$

It then applies the following four steps, which we will call a *Grover iteration*, repeatedly until we are done. We will calculate the exact number of times we need to apply the Grover iteration later.

The Grover iteration is:

1. Apply the oracle O_p,
2. Apply the Hadamard transform $H^{\otimes n}$
3. Apply the unitary transformation $2\,|\,0\rangle\langle 0\,| - I$,
4. Apply the Hadamard transform $H^{\otimes n}$.

The third step, $2\,|\,0\rangle\langle 0\,| - I$, is unitary and easy to implement. To see why, consider the case of two qubits. The transformation is

$$2\,|\,0\rangle\langle 0\,| - I = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}.$$

So it applies a -1 phase if the state is not $|\,0\rangle$ and a 1 phase if it is $|\,0\rangle$. To do this, you check whether it is $|\,0\rangle$, and if not you apply a -1 phase. This transformation can be constructed out of Toffoli gates, σ_x gates and σ_z gates (try this as an exercise).

Let's look at the last three steps. Let $|\,\psi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |\,i\rangle$. Then, because $H^{\otimes n}\,|\,0\rangle = |\,\psi\rangle$,

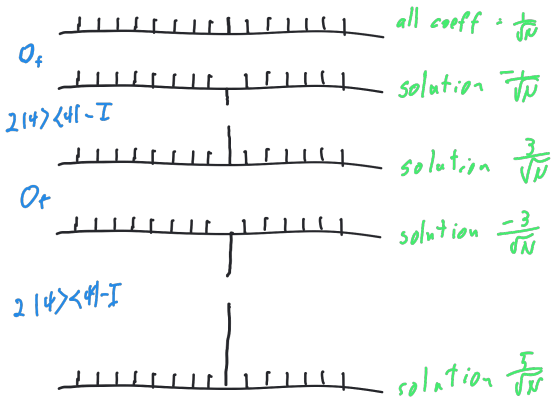
$$H^{\otimes n}(2\,|\,0\rangle\langle 0\,| - I)H^{\otimes n} = 2\,|\,\psi\rangle\langle\psi\,| - I.$$

What does $2\,|\,\psi\rangle\langle\psi\,| - I$ do? It reflects all the amplitudes around their average value. The operator $\frac{1}{N}\sum_{i=0}^{N-1} \langle i\,|$ computes the average value of the amplitude, and the operator $\sum_{i=0}^{N-1} |\,i\rangle$ applies this value to every state. And together, we have

$$\left(\sum_{i=0}^{N-1} |\,i\rangle\right)\left(\frac{1}{N}\sum_{i=0}^{N-1} \langle i\,|\right) = |\,\psi\rangle\langle\psi\,|\,.$$

Why does this work, and how many times do we have to apply the iteration? We will start by giving some intuition by explaining Grover's original analysis, and then describe a different way of analyzing the algorithm which comes up with a more accurate answer, and gives a different intuition for why it works.

The O_p steps reflect each of the amplitudes around 0, and the other three steps of the Grover iteration reflect them around the average amplitude. Let's see what happens:



The marked state starts off with amplitude $\frac{1}{\sqrt{N}}$. The first reflection around the x-axis takes it to $-\frac{1}{\sqrt{N}}$. At this point, the average will still be almost exactly $\frac{1}{\sqrt{N}}$, so the reflection around the average takes it to $\frac{3}{\sqrt{N}}$. After two more steps, it is $\frac{5}{\sqrt{N}}$. And after the kth Grover iteration, as long as the average amplitude stays around $\frac{1}{\sqrt{N}}$, the marked state will have amplitude $\frac{2k+1}{\sqrt{N}}$. So it seems reasonable that after $O(\sqrt{N})$ steps, almost all the amplitude will be in the marked state, and we will have probability nearly 1 of finding it.

This is the analysis that Grover used to discover his algorithm. The textbook uses a more accurate analysis. We now explain it. Suppose now we have M marked states.

Let $|\,\alpha\rangle$ be the equal superposition of all unmarked states:

$$|\,\alpha\rangle = \frac{1}{\sqrt{N-M}} \sum_{x \text{ unmarked}} |\,x\rangle\,,$$

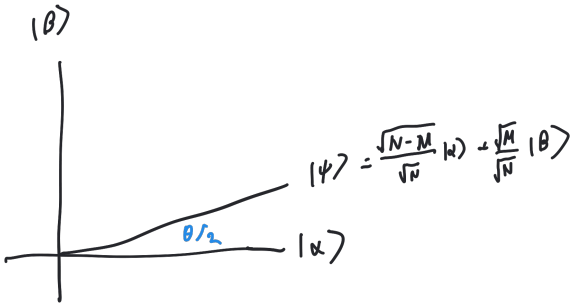
and let $|\beta\rangle$ be the equal superposition of all marked states:

$$|\,\beta\rangle = \frac{1}{\sqrt{M}} \sum_{x \text{ marked}} |\,x\rangle\,.$$

We have $|\,\psi\rangle$ is the equal superposition of all states, so

$$|\,\psi\rangle = \frac{\sqrt{N-M}}{\sqrt{N}}\,|\,\alpha\rangle + \frac{\sqrt{M}}{\sqrt{N}}\,|\,\beta\rangle\,.$$

The Grover iteration treats all marked states alike, and all unmarked states alike. It starts in a superposition of $|\,\alpha\rangle$ and $|\,\beta\rangle$, so during the algorithm, the computational state will always remain in the subspace generated by $|\,\alpha\rangle$ and $|\,\beta\rangle$. We can thus picture it as acting in a plane:



Let $\frac{\theta}{2}$ be the angle between $|\,\alpha\rangle$ and $|\,\psi\rangle$. By trigonometry,

$$\sin\left(\frac{\theta}{2}\right) = \frac{\sqrt{M}}{\sqrt{N}},$$

so $\theta \approx 2\frac{\sqrt{M}}{\sqrt{N}}$ if $M \ll N$.

What do O_p and $2\,|\,\psi\rangle\langle\psi\,| - I$ do to the state? The operation O_p takes $|\,\alpha\rangle$ to $|\,\alpha\rangle$, because it acts as the identity on unmarked states, and $|\,\beta\rangle$ to $-\,|\,\beta\rangle$, because it applies a phase of -1 to marked states. The operation $2\,|\,\psi\rangle\langle\psi\,| - I$ takes $|\,\psi\rangle$ to $|\,\psi\rangle$, because

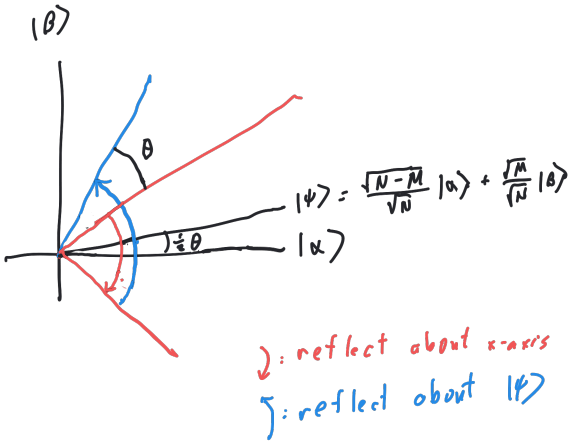
$$\left(2\,|\,\psi\rangle\langle\psi\,| - I\right)|\,\psi\rangle = 2\,|\,\psi\rangle - |\,\psi\rangle = |\,\psi\rangle\,.$$

and if $\left|\,\bar{\psi}\right\rangle$ is the state in the $|\,\alpha\rangle$, $|\,\beta\rangle$ plane orthogonal to $|\,\psi\rangle$, then

$$\left(2\,|\,\psi\rangle\langle\psi\,| - I\right)\left|\,\bar{\psi}\right\rangle = -\left|\,\bar{\psi}\right\rangle.$$

These two facts mean that $2\,|\,\psi\rangle - I$ is a reflection of the plane around the line through 0 and $|\,\psi\rangle$.

What is the product of two reflections in the plane?



If the two reflections are about lines that intersect at an angle of ϕ radians, then their product is a rotation around the intersection by an angle of 2ϕ radians. Thus, each Grover iteration rotates the state by approximately $2\frac{\theta}{2} = \theta \approx 2\frac{\sqrt{M}}{\sqrt{N}}$ radians. We would like the state to be rotated to the axis $|\,\beta\rangle$, which is at an angle of approximately $\pi/2$ from the original state $|\,\phi\rangle$. This will take roughly

$$\frac{\frac{\pi}{2}}{2\frac{\sqrt{M}}{\sqrt{N}}} \approx \frac{\pi}{4}\frac{\sqrt{N}}{\sqrt{M}},$$

So Grover's algorithm will find a marked state in approximately $\frac{\pi}{4}\frac{\sqrt{N}}{\sqrt{M}}$ iterations.

What do we do if we don't know M? If you have a vector at a random angle, and measure it in the $|\,\beta\rangle, |\,\alpha\rangle$ basis, then $\frac{1}{2}$ of the time you will get $|\,\beta\rangle$. That is, if you run Grover's algorithm for a random number of iterations, then as long as this number of iterations is sufficiently large, you have a $\frac{1}{2}$ chance of getting a marked state. Thus, one method for finding a marked state in $O(\frac{\sqrt{N}}{\sqrt{M}})$ iterations is to run it for a random number between 5 and 10 iterations, then a random number between 10 and 20 iterations, then a random number between 20 and 40 iterations, then a random number between 40 and 80, and so on. Eventually, not long after you reach roughly $\frac{\sqrt{N}}{\sqrt{M}}$ itertions, you are very likely to find a marked state, and you haven't taken more than $O(\frac{\sqrt{N}}{\sqrt{M}})$ iterations total.