※ Computation 2: Quantum Search - Grover's Algorithm

13.1 The Search Problem

We've found a few settings of query complexity where quantum computers provide an exponential improvement. A lot of these were very artificial settings, and there is a class of problems we are interested in where the best we can do is exponential right now. This is the class of NP-complete problems, so naturally the following question was raised. "Can quantum computers solve NP-complete problems?"

Example 13.1. 3-SAT(ϕ)

- **Input:** A Boolean formula ϕ in 3-CNF form consisting of n-Boolean variables: $\phi(x_1, x_2, \ldots, x_n)$.

$$\text{3-CNF: } (x_2 \vee \overline{x}_7 \vee x_3) \wedge (x_5 \vee x_7 \vee x_9) \wedge \cdots \tag{98}$$

- **Output:** Does ϕ have a satisfying assignment? That is, is there a way of setting each x_i such that ϕ evaluates to TRUE? ↳ YES.

I can "prove" to you that there is a satisfying assignment by giving you an assignment of the variables $x_1, \ldots, x_n$, which you just need to plug into the function and check for yourself.

Question 133. Let

$$\phi(x_1, x_2, x_3, x_4) = (x_2 \vee \overline{x}_3 \vee x_4) \wedge (x_1 \vee \overline{x}_2 \vee x_4) \wedge (\overline{x}_1 \vee \overline{x_3} \vee \overline{x}_4). \tag{99}$$

What is $\phi(1,1,1,1)$? What is $\phi(1,0,0,1)$? Is ϕ a YES instance of 3-SAT?

$\phi(1,1,1,1) = 1 \wedge 1 \wedge 0 = 0$ ✗ Not satisfying.

$\phi(1,0,0,1) = 1 \wedge 1 \wedge 1 = 1$ ✓ Satisfying assignment.

YES.

If I can solve 3SAT → I can solve ALL of NP.

The above example is important because not only is it in NP, it is actually a problem in a subset of NP problems called NP-complete. NP-complete problems are often referred to as "the hardest problems in NP", and are an important set of problems because we know that an efficient solution to *any* NP-complete problem would mean there is an efficient solution to *every* problem in NP! It is generally believed that it is impossible to solve NP-complete problems in polynomial time using a classical computer. We will continue the discussion in the setting of query complexity.

Per usual, we will have black-box access to the function $f : \{0,1\}^n \to \{0,1\}$ we are interested in in the form of a unitary U_f:

$$U_f |x\rangle |y\rangle = |x\rangle |y \oplus f(x)\rangle . \tag{100}$$

We are interested in deciding some property of the function. For this last setting, we are interested in the search problem:

Definition 13.2 (Search Problem)**.** Let $f : \{0,1\}^n \to \{0,1\}$. Is there an n bit string x such that $f(x) = 1$?

For 3SAT, set $f = \phi$

This problem is interesting because any decision problem (including NP-complete ones) can be formulated in this way. For example, if we have a 3-SAT instance we can set $f(x_1 \cdots x_n) = \phi(x_1, \cdots, x_n)$. To simplify the analysis, we will first assume that f either has a unique solution or no solution at all. In other words,

$$\left|\{x | f(x) = 1\}\right| = 0 \text{ or } 1. \tag{101}$$

In the case that a solution does exist, we will call the solution string a. $f(a) = 1$, a is unique.

$f(x) = 0$ for all $x \neq a$.

Question 134. What is the classical query complexity for the above problem?

2^n queries

You may be wondering if this is *too* general for analyzing NP-complete problems, but there is a widely believed hypothesis that a brute force search is the best we can do for 3SAT. In other words, we believe that the lower bound shown above for general f holds for functions that are NP-complete too. We now want to ask what kind of speed-up is possible if we use a quantum computer in this setting.

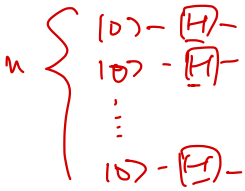
13.2 Grover's Algorithm

Let $|a\rangle$ be the standard basis state representing the value such that $f(a) = 1$. We will also use the notation

$$|\psi\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle \tag{102}$$

to represent the uniform superposition state.

Question 135. How can we create the state in the above equation?

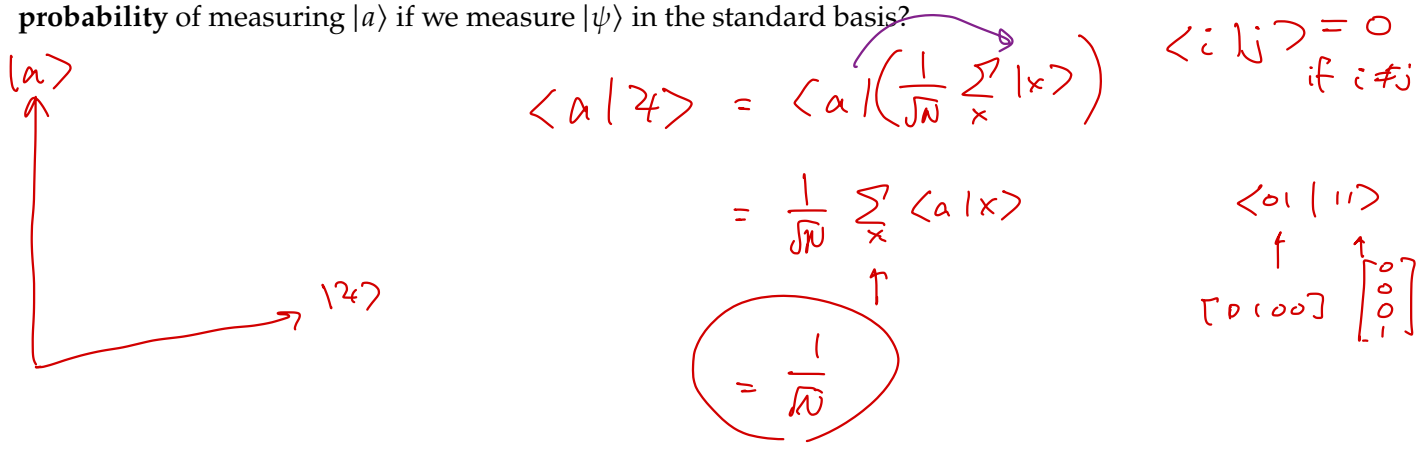


One key fact about this algorithm is that the state of our system will **always** stay in the space spanned by $|a\rangle$ and $|\psi\rangle$. In other words, for every state $|v\rangle$ that this algorithm can be in, we can express this state as a linear combination as follows:

$$|v\rangle = \alpha |a\rangle + \beta |\psi\rangle \tag{103}$$

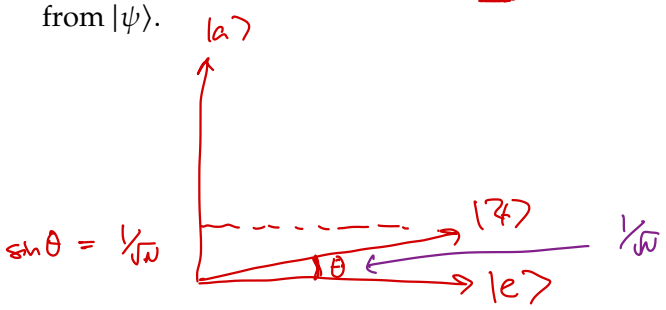
Note that $|a\rangle$ and $|\psi\rangle$ are not orthogonal, so we cannot just measure in a basis that contains both $|a\rangle$ and $|\psi\rangle$.

Question 136. What is the overlap between $|a\rangle$ and $|\psi\rangle$? Using what you found, draw the two vectors, with $|a\rangle$ as a vector pointing upward, and $|\psi\rangle$ pointing in the right direction. What is the **probability** of measuring $|a\rangle$ if we measure $|\psi\rangle$ in the standard basis?



We can create an orthonormal basis that represents the same subspace by subtracting the $|a\rangle$ component from $|\psi\rangle$.

Question 137. Create a new state $|e\rangle$ that is orthonormal to $|a\rangle$ by subtracting the $|a\rangle$ component from $|\psi\rangle$.



$$|e\rangle \propto |\psi\rangle - |a\rangle = \frac{1}{\sqrt{N-1}} \sum_{x \neq a} |x\rangle$$

if $|a\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$, $|e\rangle = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}$

The last preparation step we need is to express the angle between $|e\rangle$ and $|\psi\rangle$. For very small values of θ, it is known that

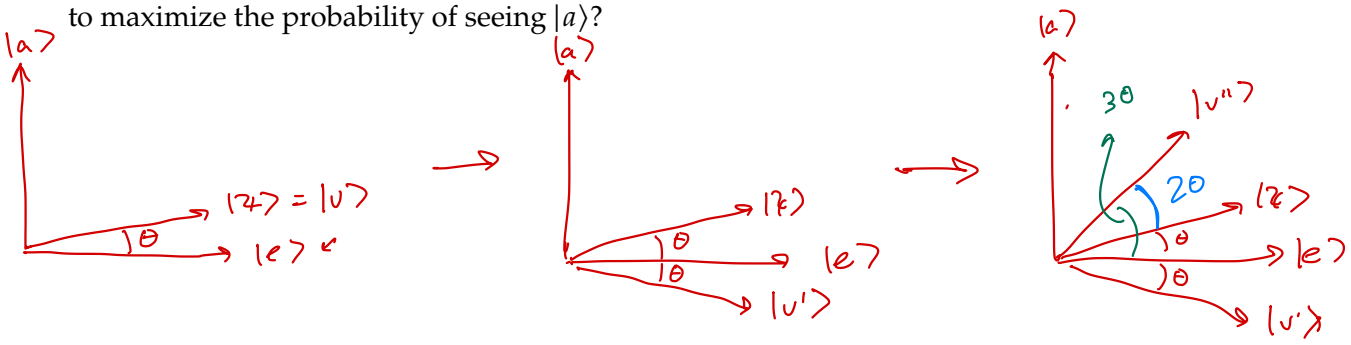
$$\sin\theta \approx \theta. \tag{104}$$

From this, we can conclude that the angle formed between $|\psi\rangle$ and $|e\rangle$ is approximately θ.

The main steps of the algorithm can be described purely geometrically. We will discuss how to implement these steps later, but for now let's build the intuition behind what the algorithm is doing. I will use $|v\rangle$ to represent the current state of the system. The algorithm starts with $|v\rangle = |\psi\rangle$. Here are the steps that we repeat:

1. Reflect $|v\rangle$ over the $|e\rangle$ axis.
2. Reflect $|v\rangle$ over $|\psi\rangle$.

Question 138. Let θ be the angle between $|v\rangle$ and $|e\rangle$ before applying the two steps above. What is the angle between the two *after* we apply the two steps? How many times do we need to do this to maximize the probability of seeing $|a\rangle$?



One iteration adds 2θ to the angle between $|e\rangle$ and $|v\rangle$.

$$(\theta + c \cdot 2\theta) = \frac{\pi}{2}$$

$$\left(\frac{1}{\sqrt{N}} + 2c \cdot \frac{1}{\sqrt{N}}\right) = \frac{\pi}{2}$$

$$(2c+1)/\sqrt{N} = \frac{\pi}{2}$$

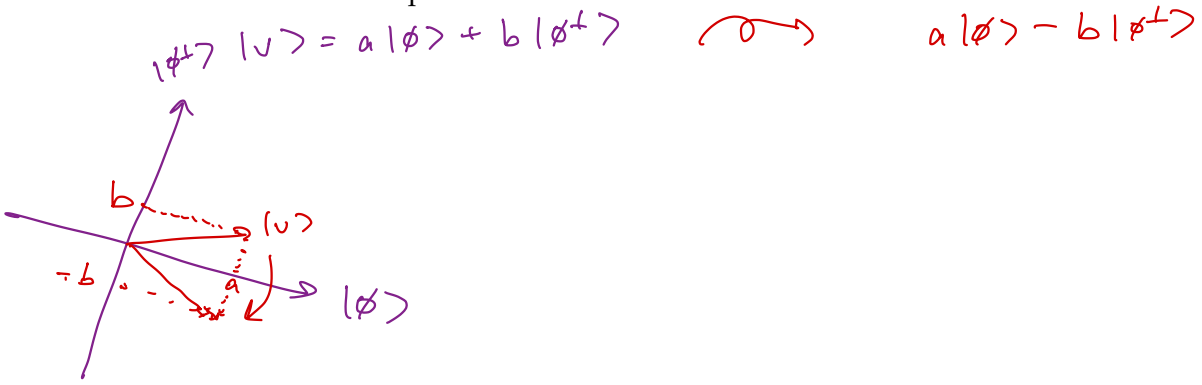
$$2c+1 = \frac{\pi}{2}\sqrt{N}$$

$$c = \frac{(\pi/2)\sqrt{N} - 1}{2} \approx O(\sqrt{2^n})$$

13.2.1 Implementing Reflections

How can we implement the reflections? To figure this out, it will help to think about how to express the geometric action of these reflections mathematically. Suppose we have a state $|v\rangle$ that we want to reflect over some other state $|\phi\rangle$.

Question 139. Write down $|v\rangle$ as a superposition of $|\phi\rangle$ and $|\phi^\perp\rangle$. Write down the state $|v'\rangle$ after the reflection is completed.



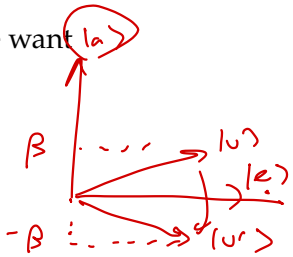
13.2.2 Reflection over $|e\rangle$

The first reflection we need to do is over the $|e\rangle$ axis. That is, we want

$$|v\rangle = \alpha\,|e\rangle + \beta\,|a\rangle \tag{105}$$

$$|v'\rangle = \alpha\,|e\rangle - \beta\,|a\rangle \tag{106}$$

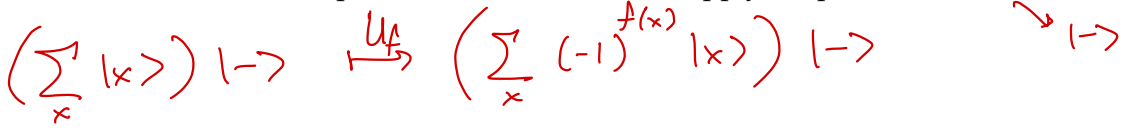
(107)

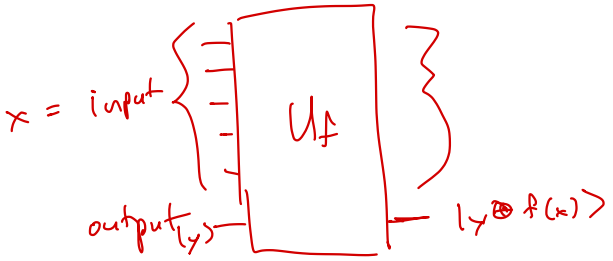


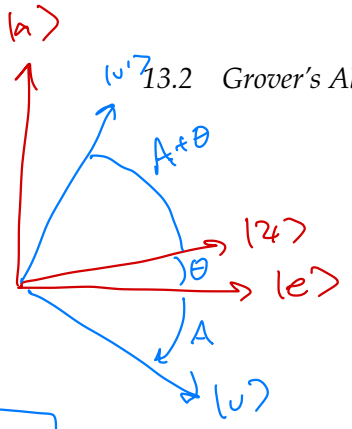
In words, we want to multiply the $|a\rangle$ state by -1, and the rest of the states by +1. Remember that a is the unique bitstring satisfying $f(a) = 1$. We can summarize this action by

$$|v\rangle = \sum_x |x\rangle \longrightarrow |v'\rangle = \sum_x (-1)^{f(x)}\,|x\rangle\,. \tag{108}$$

Question 140. We know we can perform the above operation with access to U_f. U_f acts on $n+1$ qubits. What state does this qubit need to be in for us to apply the phase of $(-1)^{f(x)}$?







13.2.3 Reflection over $|\psi\rangle$

The mapping we want to perform here is

$$|v\rangle = \alpha|\psi\rangle + \beta|\psi^\perp\rangle \tag{109}$$

$$|v'\rangle = \alpha|\psi\rangle - \beta|\psi^\perp\rangle \tag{110}$$

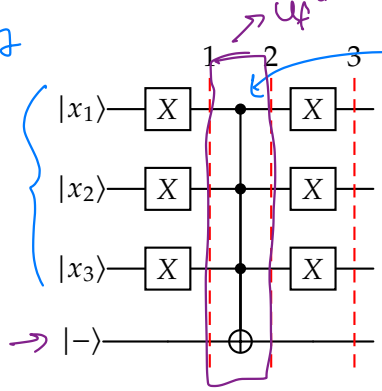
(111)

where $|\psi\rangle = \frac{1}{\sqrt{N}}\sum_x |x\rangle$.

Since $|\psi\rangle$ is generated by applying n Hadamard gates, the following is true:

$$H^{\otimes n}|0\cdots0\rangle = |\psi\rangle \iff |0\cdots0\rangle = H^{\otimes n}|\psi\rangle\,. \tag{112}$$

If we apply n Hadamard gates to $|\psi^\perp\rangle$, we get a state $|\phi\rangle$ which is a uniform superposition over all states perpendicular to $|0\cdots0\rangle$. The goal is to now apply a phase of -1 to every standard basis state which is not equal to $|0\cdots0\rangle$. I claim that the following 3 qubit circuit accomplishes this:



If input state is $|000\rangle$:

(1) $|111\rangle$

(2) $-|111\rangle|-\rangle$

(3) $-|000\rangle|-\rangle$

If input state is not $|000\rangle$:

(1) $|\bar{x}_1\bar{x}_2\bar{x}_3\rangle$

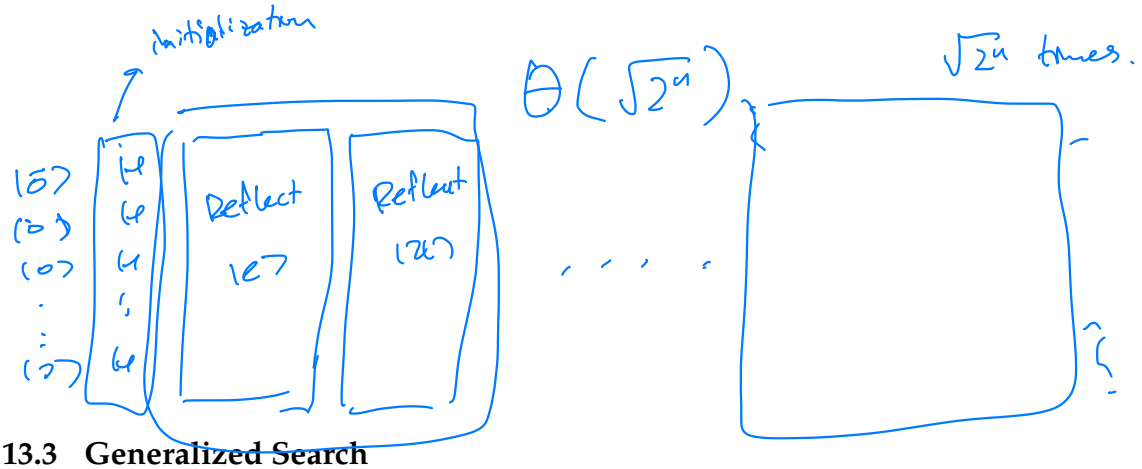
(2) $|\bar{x}_1\bar{x}_2\bar{x}_3\rangle$

(3) $|x_1x_2x_3\rangle$

$\alpha|000\rangle + \beta|000^\perp\rangle \rightarrow -\alpha|000\rangle + \beta|000^\perp\rangle$

$\alpha|000\rangle - \beta|000^\perp\rangle$

Question 141. What is the query complexity of Grover's algorithm?



13.3 Generalized Search

What about the (more likely) case where there are multiple solutions? That is,

$$\left|\{x|f(x)=1\}\right| = M > 1. \tag{113}$$

Assume (for now) that the number of solutions M is known. We would like to find any one of these solutions. We again define two orthogonal states, the uniform superpositions of the solution states and non-solution states respectively:

$$|\phi_1\rangle = \frac{1}{\sqrt{M}} \sum_{x:f(x)=1} |x\rangle \quad |\phi_0\rangle = \frac{1}{\sqrt{N-M}} \sum_{x:f(x)=0} |x\rangle\,. \tag{114}$$

Then our starting uniform superposition state can be expressed as a weighted sum of the above two states as

$$|\psi\rangle = \sqrt{\frac{N-M}{N}}\,|\phi_0\rangle + \sqrt{\frac{M}{N}}\,|\phi_1\rangle\,. \tag{115}$$

Since we know M, we can select the number of iterations c such that

$$(2c+1)\theta \approx \frac{\pi}{2} \tag{116}$$

in the same way we did in the previous section. In the case where $M << N$, $\theta \approx \sqrt{\frac{M}{N}}$, so the total number of iterations required is $O\left(\sqrt{\frac{M}{N}}\right)$.