

## Searching an unordered database

In searching for a needle in a haystack, one might hope that it pays to be systematic. Unfortunately, it does not.

Let  $f(x)$  be a function whose argument is an integer  $0 \leq x \leq N - 1$ , and which returns 1 for exactly one value  $x_1$ ; for all other values of  $x$ , it returns zero. We can think of this function as being a database query, with the numbers  $x$  labeling record numbers or memory locations; we are searching for a particular record, and the function  $f$  tells us if we have found it.

From the point of view of computation, we consider  $f$  to be an oracle which can be repeatedly queried. How many queries are necessary before the value  $x_1$  is found?

It is easy to see that an average of  $N/2$  queries will be needed to find the “marked” record; and that there is no better algorithm than to just try one value of  $x$  after another until the desired location is found.

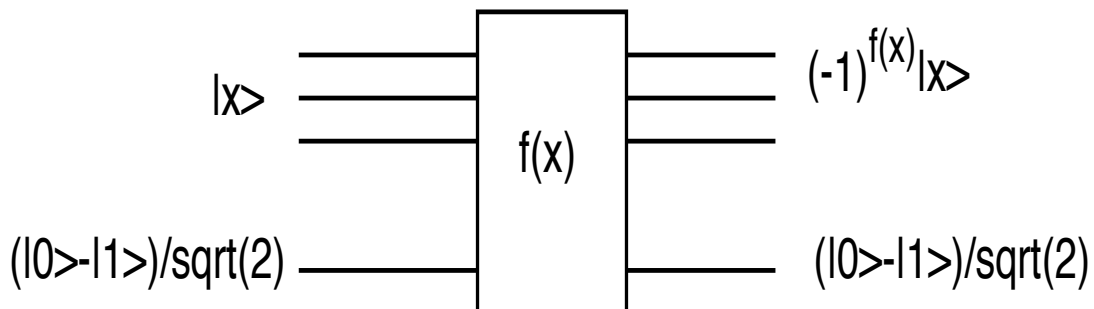
Because the problem has so little structure, every instance of the problem with the same value of  $N$  is equally difficult; and the order in which the queries are made is irrelevant. After a query, there is a probability of  $1/N$  of finding the correct record; and if not, one is left with the same problem, with size  $N - 1$ .

Can we do better with quantum mechanics?

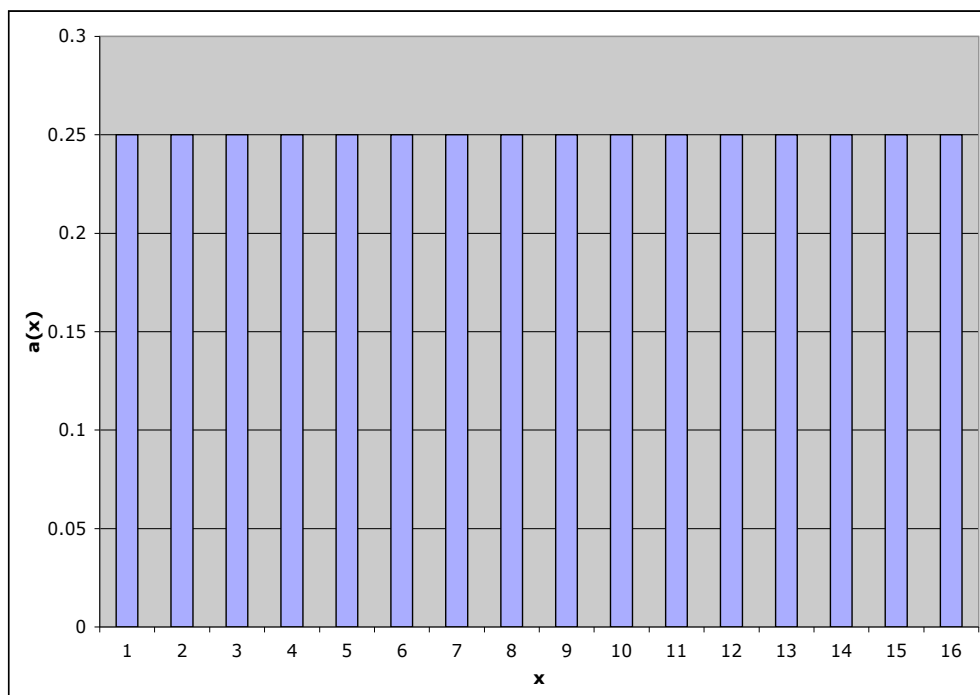
# The Grover Algorithm

Let us assume that the oracle is a unitary transformation that takes  $|x\rangle \rightarrow (-1)^{f(x)}|x\rangle$ , i.e., it flips the sign of  $|x\rangle$  if and only if  $f(x) = 1$ . Let us further assume, for the moment, that exactly one value  $x_1$  has  $f(x_1) = 1$ , and all others make  $f(x) = 0$ . (Later we will relax this assumption.)

We have seen that an oracle which takes  $|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$  can be effectively “converted” to a phase oracle by preparing  $|y\rangle$  in the state  $(|0\rangle - |1\rangle)/\sqrt{2}$ . So we will assume this phase form throughout.



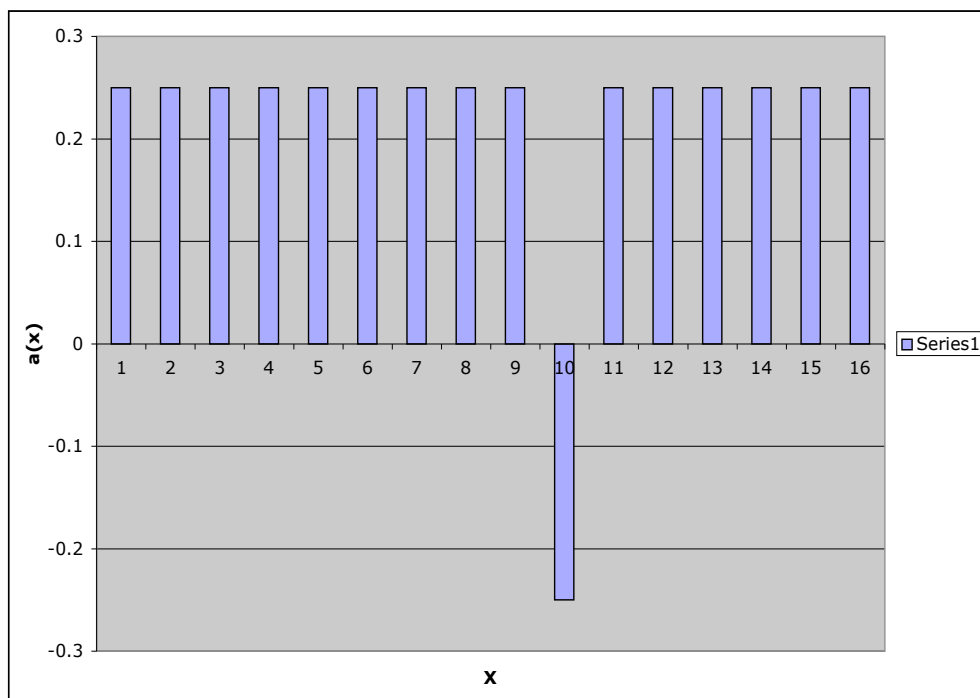
Suppose that  $N = 2^n$ , so our system is  $n$  q-bits, and we start in an evenly weighted superposition of all values  $x$ . (This can be prepared by  $n$  Hadamard gates.)



The state is then

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle.$$

If we apply the oracle, the size of the amplitudes will remain unchanged, but the amplitude of the marked record will change sign. (In the figure, the marked record is  $x = 10$ .)



The state at this point is

$$|\psi'\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle.$$

Applying the oracle again will just undo the previous application. Instead, we perform a unitary called “inversion about the mean.”

$$|\psi\rangle = \sum_x \alpha_x |x\rangle \rightarrow \sum_x (2\bar{\alpha} - \alpha_x) |x\rangle,$$

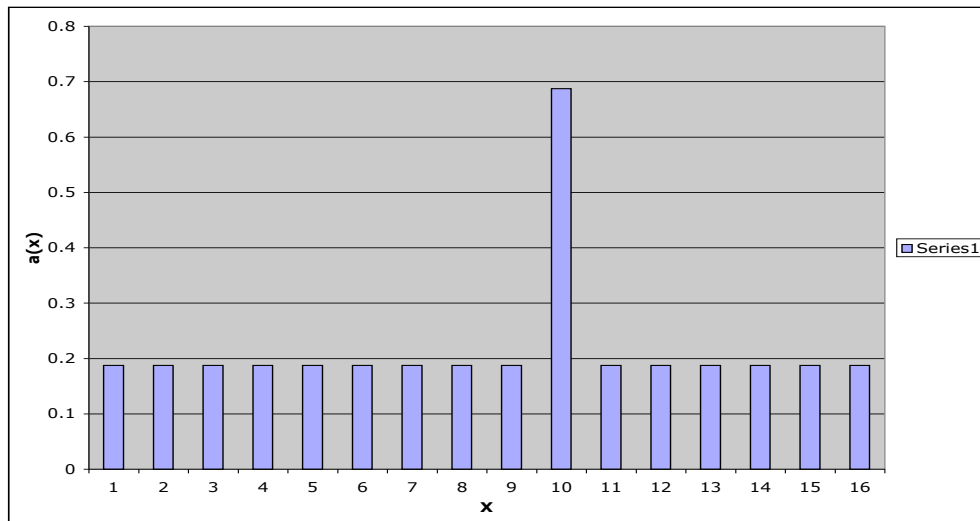
$$\bar{\alpha} = \frac{1}{N} \sum_x \alpha_x.$$

It is easy to check that this is indeed unitary.

$$\hat{U} = \begin{pmatrix} 2/N - 1 & 2/N & \cdots & 2/N \\ 2/N & 2/N - 1 & \cdots & 2/N \\ \vdots & & \ddots & \vdots \\ 2/N & 2/N & \cdots & 2/N - 1 \end{pmatrix},$$

$$\hat{U}^\dagger \hat{U} = \hat{I}.$$

The state of our q-bits now looks like this:



The probability of the marked record has grown relative to the other peaks!

By applying the oracle again, followed by “inversion about the mean,” we can make the peak grow still further; but we cannot make the peak grow without limit. After a certain number of iterations, it will reach its maximum, and then start to shrink again.

## Number of iterations

We iterate the procedure until the marked record reaches its maximum, and then measure  $x$ ; with high probability, we will find the correct value. How many iterations are needed to reach this maximum?

To answer this, we need to see that the algorithm corresponds to a rotation in a two-dimensional subspace. The initial state,  $|\psi\rangle$ , is an even superposition of all basis states. The state we are aiming for is  $|x_1\rangle$ , which is one particular basis state. Both of these states lie in the subspace spanned by the two vectors  $|x_1\rangle$  and

$$|\xi\rangle = \frac{1}{\sqrt{N-1}} \sum_{x, f(x)=0} |x\rangle.$$

These two vectors are orthogonal, and the initial state is  $|\psi\rangle = (1/\sqrt{N})|x_1\rangle + \sqrt{(N-1)/N}|\xi\rangle$ .

The important thing to notice is that the iterations of the Grover algorithm do not move you out of this space. For a state  $\alpha|x_1\rangle + \beta|\xi\rangle$ , the oracle moves you to  $-\alpha|x_1\rangle + \beta|\xi\rangle$ , which is a reflection about the state  $|\xi\rangle$ .

We can re-write the “inversion about the mean” unitary as

$$\hat{U} = 2|\psi\rangle\langle\psi| - \hat{I},$$

where once again  $|\psi\rangle$  is our evenly-weighted initial state. Since  $|\psi\rangle$  lies in the subspace, it is obvious that  $\hat{U}$  also leaves states in this subspace. In fact,  $\hat{U}$  is also a reflection, this one about the state  $|\psi\rangle$ . The product of two reflections is a rotation. Since the rotation must be independent of the state, clearly each rotation is by a constant amount. We want to rotate the state until it is as close as possible to  $|x_1\rangle$ .

Let us define the angle  $0 < \theta < \pi/2$  such that

$$\cos(\theta/2) = \sqrt{\frac{N-1}{N}},$$

so that our initial state can be written

$$|\psi\rangle = \sin(\theta/2)|x_1\rangle + \cos(\theta/2)|\xi\rangle.$$

After doing one iteration of the algorithm, the state is

$$\sin(3\theta/2)|x_1\rangle + \cos(3\theta/2)|\xi\rangle,$$

and after  $k$  iterations it is

$$\sin\left(\frac{2k+1}{2}\theta\right)|x_1\rangle + \cos\left(\frac{2k+1}{2}\theta\right)|\xi\rangle.$$

Each iteration rotates the state by  $\theta$ ; so we want to iterate until  $(2k+1)\theta \approx \pi$ .

Since  $\theta$  is small,  $\sin(\theta/2) = \sqrt{1/N} \approx \theta/2$ . Finding the marked record requires a number of steps

$$k \approx (\pi/4)\sqrt{N}.$$

By contrast with classical search, which takes  $O(N)$  queries, the quantum algorithm requires a number of oracle queries of  $O(\sqrt{N})$ .

This is not an exponential gain in speed, so in one way it is less impressive than the factoring algorithm. On the other hand, this algorithm is *provably* better than the best classical algorithm; in the case of factoring, there is no proof that the best known algorithm is really optimal.

## Searches with multiple targets

Suppose that instead of a single marked record there are  $M$  marked records, that is,  $M$  values of  $x$  for which  $f(x) = 1$ . In general, we assume  $M \ll N$ , but it need not be  $O(1)$ . For the present, we also assume that the number  $M$  is known. How does this change the search algorithm?

In fact, the procedure is exactly the same. The main difference is in the number of iterations and the final resulting state. The procedure still keeps the state within a two-dimensional subspace; in this case, the two states are

$$|\chi\rangle = \frac{1}{\sqrt{M}} \sum_{x, f(x)=1} |x\rangle,$$
$$|\xi\rangle = \frac{1}{\sqrt{N-M}} \sum_{x, f(x)=0} |x\rangle.$$

The initial state is

$$|\psi\rangle = \sqrt{\frac{M}{N}}|\chi\rangle + \sqrt{\frac{N-M}{N}}|\xi\rangle.$$

We define the angle  $0 < \theta < \pi/2$  by

$$\cos(\theta/2) = \sqrt{(N-M)/N},$$

and each step of the procedure rotates the step by  $\theta$ , just as before. The number of iterations to maximize the probability of a successful measurement is

$$k \approx \frac{\pi}{4} \sqrt{\frac{N}{M}}.$$

At the end of the process, one measures  $x$  and with high probability finds a number  $x$  for which  $f(x) = 1$ , chosen at random. (Note, however, that if  $M \geq N/2$ , the number of iterations needed actually *increases* with  $M$ , which is a bit strange. Clearly, in that limit classical searching works perfectly well.)

## Building the circuit

One part of the circuit is applying the oracle, which we already know how to do. What about this strange “inversion about the mean” unitary?

Notice that if we apply Hadamards to all  $n$  bits, the state transforms to

$$\frac{1}{\sqrt{2^n}} \sum_x \alpha_x |x\rangle \rightarrow \sum_{x,y} (-1)^{x \cdot y} \alpha_x |y\rangle,$$

where the Boolean dot product is

$$x \cdot y = (x_0 \& y_0) \oplus (x_1 \& y_1) \oplus \dots \oplus (x_{n-1} \& y_{n-1}).$$

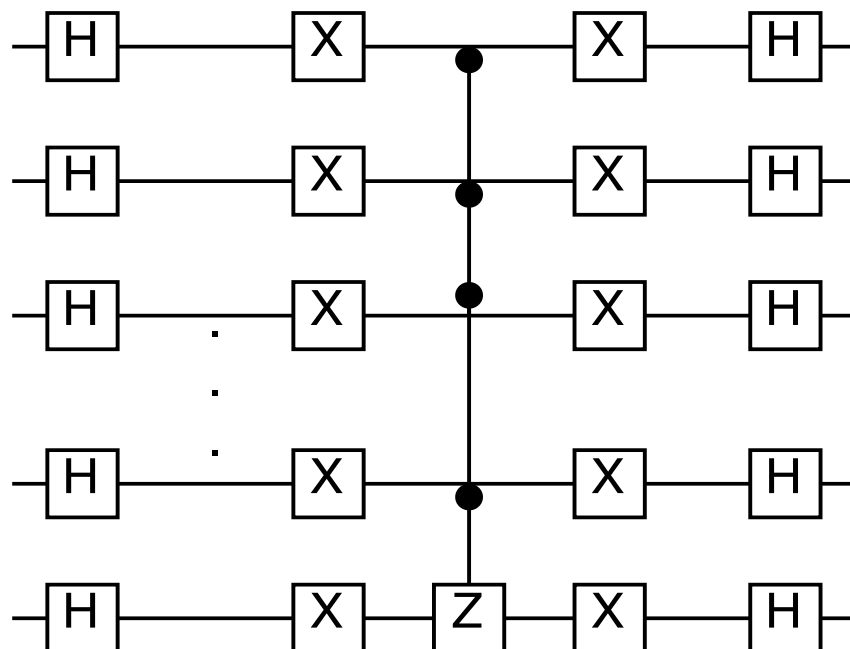
The important thing to note is that for  $y = 0$ , the new amplitude is

$$\alpha'_0 = \frac{1}{\sqrt{2^n}} \sum_x \alpha_x = \bar{\alpha}.$$

This means that “inversion about the mean” is equivalent to the following procedure:

1. Apply Hadamards to all bits.
2. Flip the sign of the  $|0\rangle$  state relative to all other basis states.
3. Apply Hadamards to all bits.

How do we carry out step 2? This is a controlled $^{n-1}$ - $Z$  gate, which we know how to build; the circuit uses  $O(n)$  gates:



## Quantum counting

We have seen that the search algorithm still works if there are  $M$  marked records. However, that demonstration required us to know the value of  $M$ , which in general we will not.

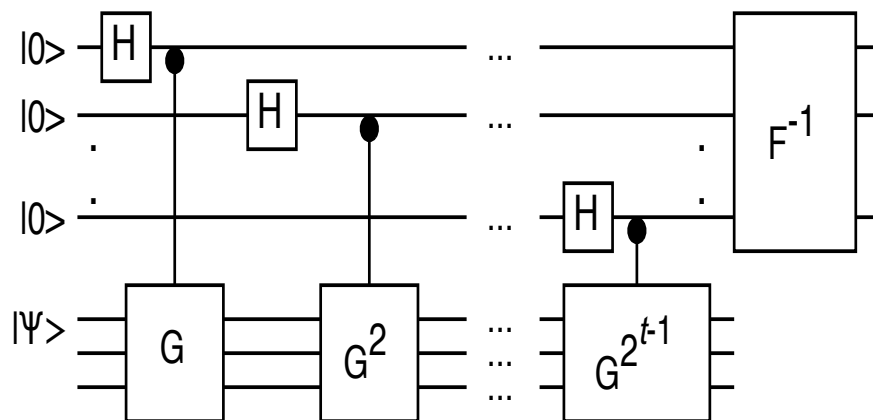
Fortunately, there is also an algorithm to estimate the number of marked records  $M$ , which also takes a time  $O(\sqrt{N})$ . This algorithm is based on our good old friend, the phase estimation algorithm.

Let us define the unitary operator  $\hat{G} = \hat{U}\hat{O}$ , where  $\hat{U}$  is inversion about the mean and  $\hat{O}$  is the oracle. (We call this operator a “Grover iteration.”) We have seen that if the q-bits start in state  $|\psi\rangle$ , they remain in a two-dimensional subspace at all subsequent steps. This subspace is spanned by two eigenvectors of  $\hat{G}$  with eigenvalues  $\exp(i\theta)$  and  $\exp(i(2\pi - \theta))$ .

As we recall, the angle  $\theta$  is such that  $\cos(\theta/2) = \sqrt{(N - M)/N}$ . Therefore, an estimate of  $\theta$  will automatically give us an estimate of  $M$ :

$$M \approx N \sin^2(\theta/2).$$

Obviously, the exact same relationship holds if we substitute  $2\pi - \theta$  for  $\theta$ . This gives us our method for estimating  $M$ .



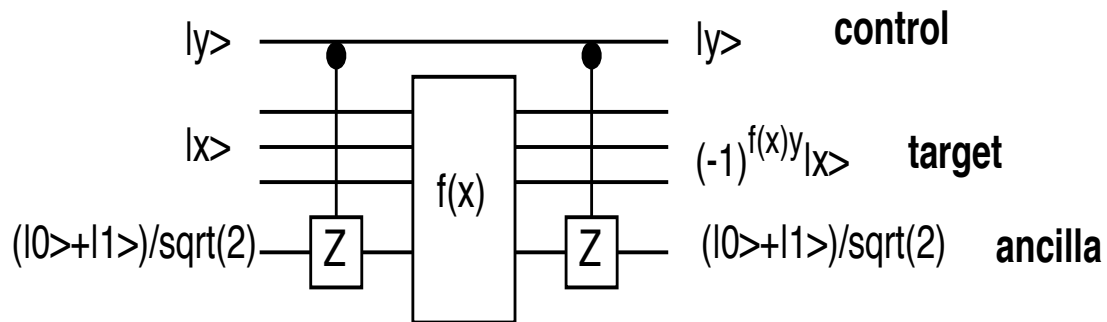
1. Prepare  $n$  target bits and  $t$  control bits in the state  $|0\rangle$ .
2. Do Hadamards on all the bits. This puts the control bits in a superposition of all  $x$  between 0 and  $2^n - 1$ , and the target bits in the state  $|\psi\rangle$  (which is a superposition of the two eigenvectors of  $\hat{G}$ ).
3. For each control bit, do a controlled- $\hat{G}^{2^m}$  from bit  $m$  onto the target bits.
4. Do an inverse Fourier transform on the control bits.
5. Measure the control bits. This will give an estimate of  $\theta$  (or  $2\pi - \theta$ ) accurate to  $\ell$  bits. Then use this to estimate  $M \approx N \sin^2(\theta/2)$ .

It is sufficient to estimate  $\theta$  to  $\ell \sim n/2$  bits of accuracy. This requires  $\sim 2^{n/2}$  controlled- $\hat{G}$  operations. We can estimate  $M$  and then search with total complexity  $O(\sqrt{N})$ .

It might seem that we have pulled a fast one here—we have assumed that we can perform controlled- $\hat{G}$  operations, but  $\hat{G}$  involves the oracle. In fact, this is not a problem.

The original oracle performed  $|x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle$ , which we converted into the phase oracle by preparing the ancilla bit in the state  $(|0\rangle - |1\rangle)/\sqrt{2}$ . We could instead prepare the ancilla in  $(|0\rangle + |1\rangle)/\sqrt{2}$ , in which case the oracle has no effect.

To make this a controlled- $\hat{O}$  operation, we sandwich the oracle call between two controlled- $Z$  gates on the ancilla bit.



If the control bit is in state 1,  $(|0\rangle + |1\rangle)/\sqrt{2} \rightarrow (|0\rangle - |1\rangle)/\sqrt{2}$  for the oracle call, and is then switched back;  $\hat{O}$  has been performed on the target bits. If the control bit is in state 0, the oracle has no effect. We have turned the oracle into a controlled- $\hat{O}$ .

## Optimality

The Grover algorithm lets one find a marked record with a number of queries  $O(\sqrt{N})$ . Is that the best that can be done? Can we bound the number of queries needed?

This is indeed possible. Let us build up to it. First, suppose that we make no calls to the oracle at all, but just put the qubits into a state  $|\phi\rangle$  and measure them. What is the probability of getting the correct answer  $x$ , averaged over all possible  $x$ ?

$$p = \frac{1}{N} \sum_x \langle \phi|x\rangle \langle x|\phi\rangle = (1/N) \langle \phi|\phi\rangle = 1/N.$$

This can do no better than making a random guess. We can try to improve things by inserting an oracle query, so our state becomes  $\hat{U}_1 \hat{O}_x |\phi\rangle$  (if the marked record is  $x$ ), where  $\hat{U}_1$  is some fixed unitary.

In order to maximize the probability of measuring the right value of  $x$ , we need the states to differ as much as possible, based on the value of  $x$ . Define

$$|\psi_k^x\rangle \equiv \hat{U}_k \hat{O}_x \hat{U}_{k-1} \hat{O}_x \cdots \hat{U}_1 \hat{O}_x |\psi\rangle,$$

$$|\psi_k\rangle \equiv \hat{U}_k \hat{U}_{k-1} \cdots \hat{U}_1 |\psi\rangle,$$

with the distance squared between them

$$D_k \equiv \sum_x \|(|\psi_k^x\rangle - |\psi_k\rangle)\|^2.$$

In order to have a good probability (say  $p > 1/2$ ) of finding the marked record, there must be some choice of  $\hat{U}_1, \dots, \hat{U}_k$  such that for almost all  $x$ ,  $|\langle x | \psi_k^x \rangle| \geq \sqrt{1/2}$ . If this is true, then it must also be true that  $D_k$  is  $\Omega(N)$ .

However, it is not difficult to show that  $D_k$  can grow only as  $O(k^2)$ .

$$\begin{aligned}
D_{k+1} &= \sum_x \|\hat{O}_x |\psi_k^x\rangle - |\psi_k\rangle\|^2 \\
&= \sum_x \|\hat{O}_x (|\psi_k^x\rangle - |\psi_k\rangle) + (\hat{O}_x - \hat{I})|\psi_k\rangle\|^2 \\
&\leq \sum_x \left( \|(|\psi_k^x\rangle - |\psi_k\rangle)\| + \|(\hat{O}_x - \hat{I})|\psi_k\rangle\| \right)^2
\end{aligned}$$

Using  $(\hat{O}_x - \hat{I}) = -2|x\rangle\langle x|$ , this becomes

$$\begin{aligned}
D_{k+1} &\leq \sum_x \left( \|(|\psi_k^x\rangle - |\psi_k\rangle)\| + 2|\langle x|\psi_k\rangle| \right)^2 \\
&\leq D_k + 4\sqrt{D_k} + 4.
\end{aligned}$$

Starting the induction with  $D_0 = 0$ , we see  $D_k \leq 4k^2$ , which implies that  $k$  must be at least  $O(\sqrt{N})$ . So the Grover algorithm is essentially optimal.

## Applications

In principle, the Grover algorithm could be used to search a database. The database could be classical, but it would have to have a quantum interface.

A much more likely application would be to speed the solution of NP-complete problems. One method of solving decision problems (such as SAT) is to try each possible solution and check if it satisfies the decision criterion. Classically, this is like searching an unordered database, and requires a time of  $O(2^n)$  for problems of size  $n$ .

By using Grover's algorithm, the time could be reduced to  $O(2^{n/2})$ , instead. This is, unfortunately, still exponential. But in practice, it could be enormously faster. More generally, we might use quantum searching to speed any program that checks many cases.