And finally, to clear the sixth column of 1's, we are going to "add" Equation (v) to Equations (i), (ii), and (vi). We get

$$\begin{array}{rl}
\text{(i)} & \langle 1000010, \mathbf{c}\rangle = 0 \\
\text{(ii)} & \langle 0010000, \mathbf{c}\rangle = 0 \\
\text{(iii)} & \langle 0100010, \mathbf{c}\rangle = 0 \\
\text{(iv)} & \langle 0001010, \mathbf{c}\rangle = 0 \\
\text{(v)} & \langle 0000001, \mathbf{c}\rangle = 0 \\
\text{(vi)} & \langle 0000000, \mathbf{c}\rangle = 0 \\
\text{(vii)} & \langle 0000100, \mathbf{c}\rangle = 0.
\end{array}$$

We can interpret these equations as

$$\begin{array}{rl}
\text{(i)} & c_1 \oplus c_6 = 0 \\
\text{(ii)} & c_3 = 0 \\
\text{(iii)} & c_2 \oplus c_6 = 0 \\
\text{(iv)} & c_4 \oplus c_6 = 0 \\
\text{(v)} & c_7 = 0 \\
\text{(vi)} & \\
\text{(vii)} & c_5 = 0.
\end{array}$$

Notice that if $c_6 = 0$, then $c_1 = c_2 = c_4 = 0$ and that if $c_6 = 1$, then $c_1 = c_2 = c_4 = 1$. Because we are certain that f is not one to one and $\mathbf{c} \neq 0000000$, we can conclude that $\mathbf{c} = 1101010$. □

Exercise 6.3.3 Solve the following linear equations in a similar manner:

$$\begin{array}{rl}
\text{(i)} & \langle 11110000, \mathbf{c}\rangle = 0 \\
\text{(ii)} & \langle 01101001, \mathbf{c}\rangle = 0 \\
\text{(iii)} & \langle 10010110, \mathbf{c}\rangle = 0 \\
\text{(iv)} & \langle 00111100, \mathbf{c}\rangle = 0 \\
\text{(v)} & \langle 11111111, \mathbf{c}\rangle = 0 \\
\text{(vi)} & \langle 11000011, \mathbf{c}\rangle = 0 \\
\text{(vii)} & \langle 10001110, \mathbf{c}\rangle = 0 \\
\text{(viii)} & \langle 01110001, \mathbf{c}\rangle = 0.
\end{array}$$

(Hint: The answer is $\mathbf{c} = 10011001$.) ■

In conclusion, for a given periodic f, we can find the period $\mathbf{c}$ in n function evaluations. This is in contrast to the $2^{n-1} + 1$ needed with the classical algorithm.

Reader Tip. We shall see this concept of finding the period of a function in Section 6.5 when we present Shor's algorithm. ♡

6.4 GROVER'S SEARCH ALGORITHM

How do you find a needle in a haystack? You look at each piece of hay separately and check each one to see if it is the desired needle. That is not very efficient.

The computer science version of this problem is about unordered arrays instead of haystacks. Given an unordered array of m elements, find a particular element. Classically, in the worst case, this takes m queries. On average, we will find the desired element in $m/2$ queries. Can we do better?

Lov Grover's search algorithm does the job in $\sqrt{m}$ queries. Although this is not the exponential speedup of the Deutsch–Jozsa algorithm and Simon's algorithm, it is still very good. Grover's algorithm has many applications to database theory and other areas.

Because, over the past few sections, we have become quite adept at functions, let us look at the search problem from the point of view of functions. Imagine that you are given a function $f : \{0,1\}^n \longrightarrow \{0,1\}$ and you are assured that there exists exactly one binary string $\mathbf{x_0}$ such that

$$f(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} = \mathbf{x_0}, \\ 0, & \text{if } \mathbf{x} \neq \mathbf{x_0}. \end{cases} \tag{6.100}$$

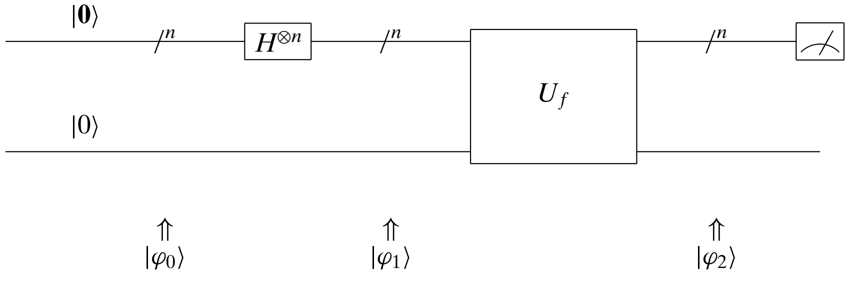
We are asked to find $\mathbf{x_0}$. Classically, in the worst case, we would have to evaluate all 2^n binary strings to find the desired $\mathbf{x_0}$. Grover's algorithm will demand only $\sqrt{2^n} = 2^{\frac{n}{2}}$ evaluations.

f will be given to us as the unitary matrix U_f that takes $|\mathbf{x}, y\rangle$ to $|\mathbf{x}, f(\mathbf{x}) \oplus y\rangle$. For example, for $n = 2$, if f is the unique function that "picks out" the binary string 10, then U_f looks like

$$\begin{array}{c|cccccccc|}
 & \mathbf{00,0} & \mathbf{00,1} & \mathbf{01,0} & \mathbf{01,1} & \mathbf{10,0} & \mathbf{10,1} & \mathbf{11,0} & \mathbf{11,1} \\
\mathbf{00,0} & 1 & & & & & & & \\
\mathbf{00,1} & & 1 & & & & & & \\
\mathbf{01,0} & & & 1 & & & & & \\
\mathbf{01,1} & & & & 1 & & & & \\
\mathbf{10,0} & & & & & & 1 & & \\
\mathbf{10,1} & & & & & 1 & & & \\
\mathbf{11,0} & & & & & & & 1 & \\
\mathbf{11,1} & & & & & & & & 1
\end{array}. \tag{6.101}$$

Exercise 6.4.1 Find the matrices that correspond to the other three functions from $\{0,1\}^2$ to $\{0,1\}$ that have exactly one element $\mathbf{x}$ with $f(\mathbf{x}) = 1$. ■

As a first try at solving this problem, we might try placing $|\mathbf{x}\rangle$ into a superposition of all possible strings and then evaluating U_f.



$|\mathbf{0}\rangle$
$/^n$
$H^{\otimes n}$
$/^n$
U_f
$/^n$
$|0\rangle$
$\Uparrow$ $|\varphi_0\rangle$
$\Uparrow$ $|\varphi_1\rangle$
$\Uparrow$ $|\varphi_2\rangle$

(6.102)

In terms of matrices this becomes

$$U_f(H^{\otimes n} \otimes I)|\mathbf{0}, 0\rangle. \tag{6.103}$$

The states are

$$|\varphi_0\rangle = |\mathbf{0}, 0\rangle, \tag{6.104}$$

$$|\varphi_1\rangle = \left[\frac{\sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}\rangle}{\sqrt{2^n}}\right]|0\rangle, \tag{6.105}$$

and

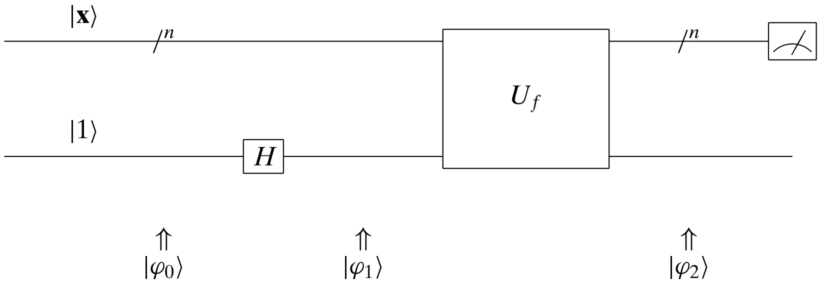
$$|\varphi_2\rangle = \frac{\sum_{\mathbf{x}\in\{0,1\}^n} |\mathbf{x}, f(\mathbf{x})\rangle}{\sqrt{2^n}}. \tag{6.106}$$

Measuring the top qubits will, with equal probability, give one of the 2^n binary strings. Measuring the bottom qubit will give $|0\rangle$ with probability $\frac{2^n-1}{2^n}$, and $|1\rangle$ with probability $\frac{1}{2^n}$. If you are lucky enough to measure $|1\rangle$ on the bottom qubit, then, because the top and the bottom are entangled, the top qubit will have the correct answer. However, it is improbable that you will be so lucky. We need something new.

Grover's search algorithm uses two tricks. The first, called **phase inversion**, changes the phase of the desired state. It works as follows. Take U_f and place the bottom qubit in the superposition

$$\frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{6.107}$$

state. This is similar to quantum circuit (6.21). For an arbitrary $\mathbf{x}$, this looks like


$|\mathbf{x}\rangle$
$/^n$
$|1\rangle$
H
U_f
$/^n$
$\Uparrow$ $|\varphi_0\rangle$
$\Uparrow$ $|\varphi_1\rangle$
$\Uparrow$ $|\varphi_2\rangle$

(6.108)

In terms of matrices this becomes

$$U_f(I_n \otimes H)|\mathbf{x}, 1\rangle. \tag{6.109}$$

Because both U_f and H are unitary operations, it is obvious that phase inversion is a unitary operation.

The states are

$$|\varphi_0\rangle = |\mathbf{x}, 1\rangle, \tag{6.110}$$

$$|\varphi_1\rangle = |\mathbf{x}\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] = \left[\frac{|\mathbf{x}, 0\rangle - |\mathbf{x}, 1\rangle}{\sqrt{2}}\right], \tag{6.111}$$

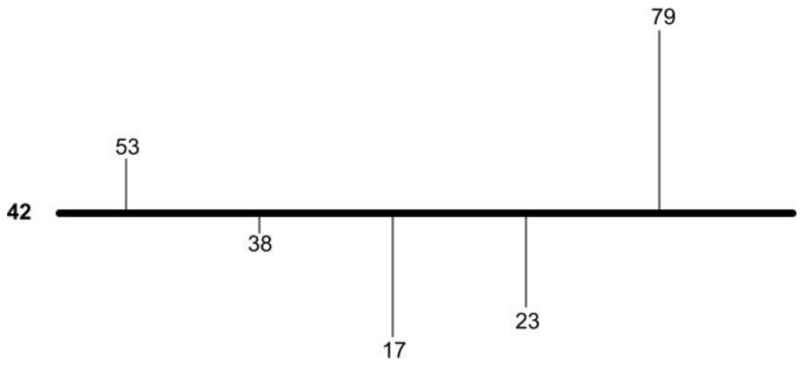


Figure 6.1. Five numbers and their average

and

$$|\varphi_2\rangle = |\mathbf{x}\rangle \left[\frac{|f(\mathbf{x}) \oplus 0\rangle - |f(\mathbf{x}) \oplus 1\rangle}{\sqrt{2}} \right] = |\mathbf{x}\rangle \left[\frac{|f(\mathbf{x})\rangle - |\overline{f(\mathbf{x})}\rangle}{\sqrt{2}} \right]. \tag{6.112}$$

Remembering that $a - b = (-1)(b - a)$, we may write

$$|\varphi_2\rangle = (-1)^{f(\mathbf{x})}|\mathbf{x}\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] = \begin{cases} -1|\mathbf{x}\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right], & \text{if } \mathbf{x} = \mathbf{x_0}, \\ +1|\mathbf{x}\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right], & \text{if } \mathbf{x} \neq \mathbf{x_0}. \end{cases} \tag{6.113}$$

How does this unitary operation act on states? If $|\mathbf{x}\rangle$ starts off in a equal superposition of four different states, i.e., $\left[\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right]^T$, and f chooses the string “10,” then after performing a phase inversion, the state looks like $\left[\frac{1}{2}, \frac{1}{2}, -\frac{1}{2}, \frac{1}{2}\right]^T$. Measuring $|\mathbf{x}\rangle$ does not give any information: both $|\frac{1}{2}|^2$ and $|-\frac{1}{2}|^2$ are equal to $+\frac{1}{4}$. Changing the phase from positive to negative separates the phases, but does not separate them enough. We need something else.

What is needed is a way of boosting the phase separation of the desired binary string from the other binary strings. The second trick is called **inversion about the mean** or **inversion about the average**. This is a way of boosting the separation of the phases. A small example will be helpful. Consider a sequence of integers: $53, 38, 17, 23$, and 79. The average of these numbers is $a = 42$. We might picture these numbers as in Figure 6.1.

The average is the number such that the sum of the lengths of the lines above the average is the same as the sum of the lengths of the lines below. Suppose we wanted to change the sequence so that each element of the original sequence above the average would be the same distance from the average but below. Furthermore, each element of the original sequence below the average would be the same distance from the average but above. In other words, we are inverting each element around the average. For example, the first number, 53 is $a - 53 = -11$ units away from the average. We must add $a = 42$ to -11 and get $a + (a - 53) = 31$. The second element of the original sequence, 38, is $a - 38 = 4$ units below the average and will go to

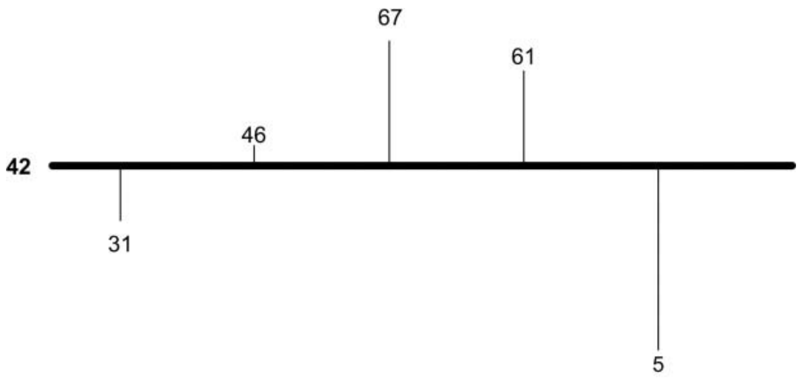


Figure 6.2. After an inversion about the mean.

$a + (a - 38) = 46$. In general, we shall change each element v to

$$v' = a + (a - v) \tag{6.114}$$

or

$$v' = -v + 2a. \tag{6.115}$$

The above sequence becomes 31, 46, 67, 61, and 5. Notice that the average of this sequence remains 42 as in Figure 6.2.

Exercise 6.4.2 Consider the following number: 5, 38, 62, 58, 21, and 35. Invert these numbers around their mean. ■

Let us write this in terms of matrices. Rather than writing the numbers as a sequence, consider a vector $V = [53, 38, 17, 23, 79]^T$. Now consider the matrix

$$A = \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}. \tag{6.116}$$

It is easy to see that A is a matrix that finds the average of a sequence:

$$AV = [42, 42, 42, 42, 42]^T. \tag{6.117}$$

In terms of matrices, the formula $v' = -v + 2a$ becomes

$$V' = -V + 2AV = (-I + 2A)V. \tag{6.118}$$

Let us calculate

$$(-I+2A) = \begin{bmatrix} (-1+\frac{2}{5}) & \frac{2}{5} & \frac{2}{5} & \frac{2}{5} & \frac{2}{5} \\ \frac{2}{5} & (-1+\frac{2}{5}) & \frac{2}{5} & \frac{2}{5} & \frac{2}{5} \\ \frac{2}{5} & \frac{2}{5} & (-1+\frac{2}{5}) & \frac{2}{5} & \frac{2}{5} \\ \frac{2}{5} & \frac{2}{5} & \frac{2}{5} & (-1+\frac{2}{5}) & \frac{2}{5} \\ \frac{2}{5} & \frac{2}{5} & \frac{2}{5} & \frac{2}{5} & (-1+\frac{2}{5}) \end{bmatrix}. \tag{6.119}$$

And, as expected,

$$(-I+2A)V = V', \tag{6.120}$$

or in our case,

$$(-I+2A)[53, 38, 17, 23, 79]^T = [31, 46, 67, 61, 5]^T. \tag{6.121}$$

Let us generalize: rather than dealing with five numbers, let us deal with 2^n numbers. Given n qubits, there are 2^n possible states. A state is a 2^n vector. Consider the following 2^n-by-2^n matrix:

$$A = \begin{bmatrix} \frac{1}{2^n} & \frac{1}{2^n} & \cdots & \frac{1}{2^n} \\ \frac{1}{2^n} & \frac{1}{2^n} & \cdots & \frac{1}{2^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{2^n} & \frac{1}{2^n} & \cdots & \frac{1}{2^n} \end{bmatrix}. \tag{6.122}$$

Exercise 6.4.3 Prove that $A^2 = A$. ■

Multiplying any state by A will give a state where each amplitude will be the average of all the amplitudes. Building on this, we form the following 2^n-by-2^n matrix

$$-I+2A = \begin{bmatrix} -1+\frac{2}{2^n} & \frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \frac{2}{2^n} & -1+\frac{2}{2^n} & \cdots & \frac{2}{2^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{2}{2^n} & \frac{2}{2^n} & \cdots & -1+\frac{2}{2^n} \end{bmatrix}. \tag{6.123}$$

Multiplying a state by $-I+2A$ will invert amplitudes about the mean.

We must show that $-I+2A$ is a unitary matrix. First, observe that the adjoint of $-I+2A$ is itself. Then, using the properties of matrix multiplication and realizing that matrices act very much like polynomials, we have

$$\begin{aligned}(-I+2A)\star(-I+2A) &= +I-2A-2A+4A^2 \\ &= I-4A+4A^2 = I-4A+4A = I,\end{aligned} \tag{6.124}$$

where the first equality is from distributivity of matrix multiplication, the second equality comes from combining like terms, and the third equality is from the fact that $A^2 = A$. We conclude that $(-I + 2A)$ is a unitary operation and acts on states by inverting the numbers about the mean.

When considered separately, phase inversion and inversion about the mean are each innocuous operations. However, when combined, they are a very powerful operation that separates the amplitude of the desired state from those of all the other states.

Example 6.4.1 Let us do an example that shows how both these techniques work together. Consider the vector

$$[10, 10, 10, 10, 10]^T. \tag{6.125}$$

We are always going to perform a phase inversion to the fourth of the five numbers. There is no difference between the fourth number and all the other numbers. We start by doing a phase inversion to the fourth number and get

$$[10, 10, 10, -10, 10]^T. \tag{6.126}$$

The average of these five numbers is $a = 6$. Calculating the inversion about the mean we get

$$-v + 2a = -10 + (2 \times 6) = 2 \tag{6.127}$$

and

$$-v + 2a = 10 + (2 \times 6) = 22. \tag{6.128}$$

Thus, our five numbers become

$$[2, 2, 2, 22, 2]^T. \tag{6.129}$$

The difference between the fourth element and all the others is $22 - 2 = 20$.

Let us do these two operations again to our five numbers. Another phase inversion on the fourth element gives us

$$[2, 2, 2, -22, 2]^T. \tag{6.130}$$

The average of these numbers is $a = -2.8$. Calculating the inversion about the mean we get

$$-v + 2a = -2 + (2 \times -2.8) = -7.6 \tag{6.131}$$

and

$$-v + 2a = 22 + (2 \times -2.8) = 16.4. \tag{6.132}$$

Hence, our five numbers become

$$[-7.6, -7.6, -7.6, 16.4, -7.6]^T. \tag{6.133}$$

The difference between the fourth element and all the others is $16.4 + 7.6 = 24$. We have further separated the numbers. This was all done with unitary operations. □

Exercise 6.4.4 Do the two operations again on this sequence of five numbers. Did our results improve? ■

How many times should these operations be done? $\sqrt{2^n}$ times. If you do it more than that, the process will "overcook" the numbers. The proof that $\sqrt{2^n}$ times is needed is beyond this text. Suffice it to say that the proof actually uses some very pretty geometry (well worth looking into!).

We are ready to state Grover's algorithm:

Step 1. Start with a state $|\mathbf{0}\rangle$

Step 2. Apply $H^{\otimes n}$

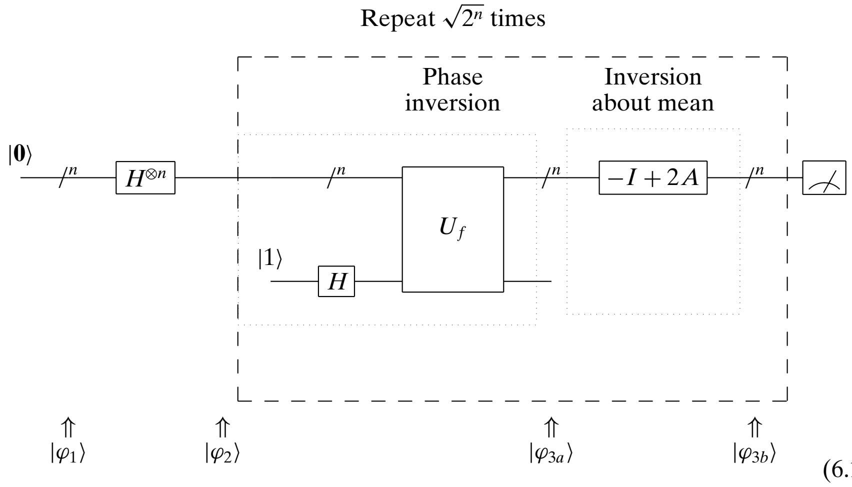
Step 3. Repeat $\sqrt{2^n}$ times:

Step 3a. Apply the phase inversion operation: $U_f(I \otimes H)$

Step 3b. Apply the inversion about the mean operation: $-I + 2A$

Step 4. Measure the qubits.

We might view this algorithm as



(6.134)

Example 6.4.2 Let us look at an example of an execution of this algorithm. Let f be a function that picks out the string "101." The states after each step will be

$$|\varphi_1\rangle = \begin{array}{c} \\ \end{array}\begin{bmatrix} \mathbf{000} & \mathbf{001} & \mathbf{010} & \mathbf{011} & \mathbf{100} & \mathbf{101} & \mathbf{110} & \mathbf{111} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T, \tag{6.135}$$

$$|\varphi_2\rangle = \begin{array}{c} \begin{array}{cccccccc} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \end{array} \\ \left[\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right]^T \end{array}, \tag{6.136}$$

$$|\varphi_{3a}\rangle = \begin{array}{c} \begin{array}{cccccccc} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \end{array} \\ \left[\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, -\frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}, \frac{1}{\sqrt{8}}\right]^T \end{array}. \tag{6.137}$$

The average of these numbers is

$$a = \frac{7 * \frac{1}{\sqrt{8}} - \frac{1}{\sqrt{8}}}{8} = \frac{\frac{6}{\sqrt{8}}}{8} = \frac{3}{4\sqrt{8}}. \tag{6.138}$$

Calculating the inversion about the mean we have

$$-v + 2a = -\frac{1}{\sqrt{8}} + \left(2 \times \frac{3}{4\sqrt{8}}\right) = \frac{1}{2\sqrt{8}} \tag{6.139}$$

and

$$-v + 2a = \frac{1}{\sqrt{8}} + \left(2 \times \frac{3}{4\sqrt{8}}\right) = \frac{5}{2\sqrt{8}}. \tag{6.140}$$

Thus, we have

$$|\varphi_{3b}\rangle = \begin{array}{c} \begin{array}{cccccccc} \mathbf{000} & \mathbf{001} & \mathbf{010} & \mathbf{011} & \mathbf{100} & \mathbf{101} & \mathbf{110} & \mathbf{111} \end{array} \\ \left[\frac{1}{2\sqrt{8}}, \frac{1}{2\sqrt{8}}, \frac{1}{2\sqrt{8}}, \frac{1}{2\sqrt{8}}, \frac{1}{2\sqrt{8}}, \frac{5}{2\sqrt{8}}, \frac{1}{2\sqrt{8}}, \frac{1}{2\sqrt{8}}\right]^T \end{array}. \tag{6.141}$$

A phase inversion will give us

$$|\varphi_{3a}\rangle = \begin{array}{c} \begin{array}{cccccccc} \mathbf{000} & \mathbf{001} & \mathbf{010} & \mathbf{011} & \mathbf{100} & \mathbf{101} & \mathbf{110} & \mathbf{111} \end{array} \\ \left[\frac{1}{2\sqrt{8}}, \frac{1}{2\sqrt{8}}, \frac{1}{2\sqrt{8}}, \frac{1}{2\sqrt{8}}, \frac{1}{2\sqrt{8}}, -\frac{5}{2\sqrt{8}}, \frac{1}{2\sqrt{8}}, \frac{1}{2\sqrt{8}}\right]^T \end{array}. \tag{6.142}$$

The average of these numbers is

$$a = \frac{7 * \frac{1}{2\sqrt{8}} - \frac{5}{2\sqrt{8}}}{8} = \frac{1}{8\sqrt{8}}. \tag{6.143}$$

Calculating for the inversion about the mean we have

$$-v + 2a = -\frac{1}{2\sqrt{8}} + \left(2 \times \frac{1}{8\sqrt{8}}\right) = -\frac{1}{4\sqrt{8}} \tag{6.144}$$

and

$$-v + 2a = \frac{5}{2\sqrt{8}} + \left(2 \times \frac{1}{8\sqrt{8}}\right) = \frac{11}{4\sqrt{8}}. \tag{6.145}$$

Hence, we have

$$|\varphi_{3b}\rangle = \begin{array}{c} \begin{array}{cccccccc} \mathbf{000} & \mathbf{001} & \mathbf{010} & \mathbf{011} & \mathbf{100} & \mathbf{101} & \mathbf{110} & \mathbf{111} \end{array}^{\;T} \\ \left[\begin{array}{cccccccc} \frac{-1}{4\sqrt{8}}, & \frac{-1}{4\sqrt{8}}, & \frac{-1}{4\sqrt{8}}, & \frac{-1}{4\sqrt{8}}, & \frac{-1}{4\sqrt{8}}, & \frac{11}{4\sqrt{8}}, & \frac{-1}{4\sqrt{8}}, & \frac{-1}{4\sqrt{8}} \end{array} \right] \end{array}. \tag{6.146}$$

For the record, $\frac{11}{4\sqrt{8}} = 0.97227$ and $\frac{-1}{4\sqrt{8}} = -0.08839$. Squaring the numbers gives us the probability of measuring those numbers. When we measure the state in Step 4, we will most likely get the state

$$|\varphi_4\rangle = \begin{array}{c} \begin{array}{cccccccc} \mathbf{000} & \mathbf{001} & \mathbf{010} & \mathbf{011} & \mathbf{100} & \mathbf{101} & \mathbf{110} & \mathbf{111} \end{array}^{\;T} \\ \left[\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right] \end{array}, \tag{6.147}$$

which is exactly what we wanted. □

Exercise 6.4.5 Do a similar analysis for the case where $n = 4$ and f chooses the "1101" string. ■

A classical algorithm will search an unordered array of size n in n steps. Grover's algorithm will take time $\sqrt{n}$. This is what is referred to as a quadratic speedup. Although this is very good, it is not the holy grail of computer science: an exponential speedup. In the next section we shall meet an algorithm that does have such a speedup.

What if we relax the requirements that there be only one needle in the haystack? Let us assume that there are t objects that we are looking for (with $t < \frac{2^n}{2}$). Grover's algorithm still works, but now one must go through the loop $\sqrt{\frac{2^n}{t}}$ times. There are many other types of generalizations and assorted changes that one can do with Grover's algorithm. Several references are given at the end of the chapter. We discuss some complexity issues with Grover's algorithm at the end of Section 8.3.

6.5 SHOR'S FACTORING ALGORITHM

The problem of factoring integers is very important. Much of the World Wide Web's security is based on the fact that it is "hard" to factor integers on classical computers. Peter Shor's amazing algorithm factors integers in polynomial time and really brought quantum computing into the limelight.

Shor's algorithm is based on the following fact: the factoring problem can be reduced to finding the period of a certain function. In Section 6.3 we learned how to find the period of a function. In this section, we employ some of those periodicity techniques to factor integers.

We shall call the number we wish to factor N. In practice, N will be a large number, perhaps hundreds of digits long. We shall work out all the calculations for the numbers 15 and 371. For exercises, we ask the reader to work with the number 247. We might as well give away the answer and tell you that the only nontrivial factors of 247 are 19 and 13.

We assume that the given N is not a prime number but is a composite number. There now exists a deterministic, polynomial algorithm that determines if N is prime