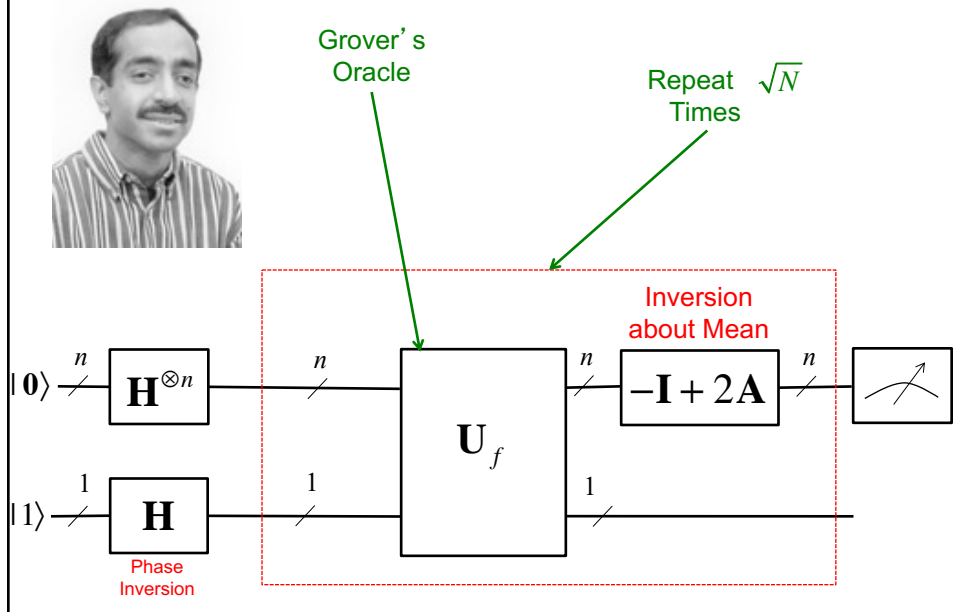


Grover Search Algorithm



1

Grover's Search Algorithm

- Method for Searching for Single Element in Among Array of N Elements
- Not Exponential Speedup Like Other Algorithms – Speedup is Quadratic $O(\sqrt{N})$
- Like Periodicity Algorithm, Method is Probabilistic
 - Requires Several Evaluations for Answer
- Cascade uses a Structure Known as Grover's Oracle
 - Yields “1” if Object Present and “0” if Not

2

Grover's Search Algorithm

- Cast Search Problem in terms of Searching for a Binary String $\mathbf{x}_0$
- We Utilize an Oracle Function of the Form:

$$f(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} = \mathbf{x}_0 \\ 0, & \mathbf{x} \neq \mathbf{x}_0 \end{cases}$$

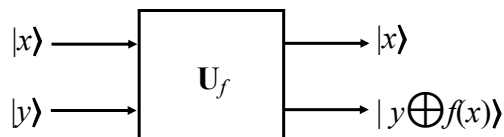
- Classical Algorithm would have to Evaluate all 2^n Strings (worst case)
- Grover's Method Requires: $\sqrt{2^n} = 2^{\frac{n}{2}}$

3

Specification of Function

- Specified as Unitary Operation that Performs the Transformation:

$$|\mathbf{x}, y\rangle \mapsto |\mathbf{x}, y \oplus f(\mathbf{x})\rangle$$



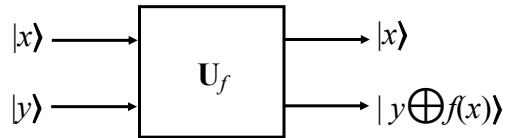
4

Function Example

- Consider $n=2$ Where $f(x)$ Detects the Bitstring

$x_0=10$:

$$|x, y\rangle \mapsto |x, y \oplus f(x)\rangle$$



$$U_f = |00,0\rangle\langle 00,0| + |00,1\rangle\langle 00,1| + |01,0\rangle\langle 01,0| + |01,1\rangle\langle 01,1| + \\ + |10,1\rangle\langle 10,0| + |10,0\rangle\langle 10,1| + |11,0\rangle\langle 11,0| + |11,1\rangle\langle 11,1|$$

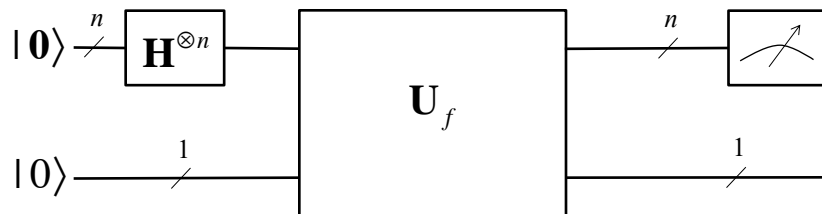
$$U_f = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- Must consider ancilla equal to both $|0\rangle$ and $|1\rangle$ since H gate is used

5

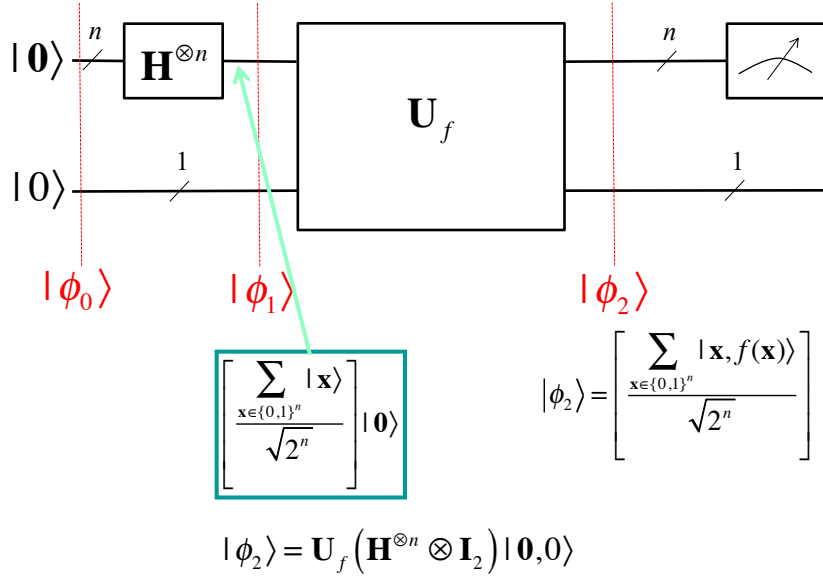
First Attempt to Solve Problem

- Use Our “Favorite Trick” of Placing Function Input into State of Superposition
- Then, Perform a Measurement



6

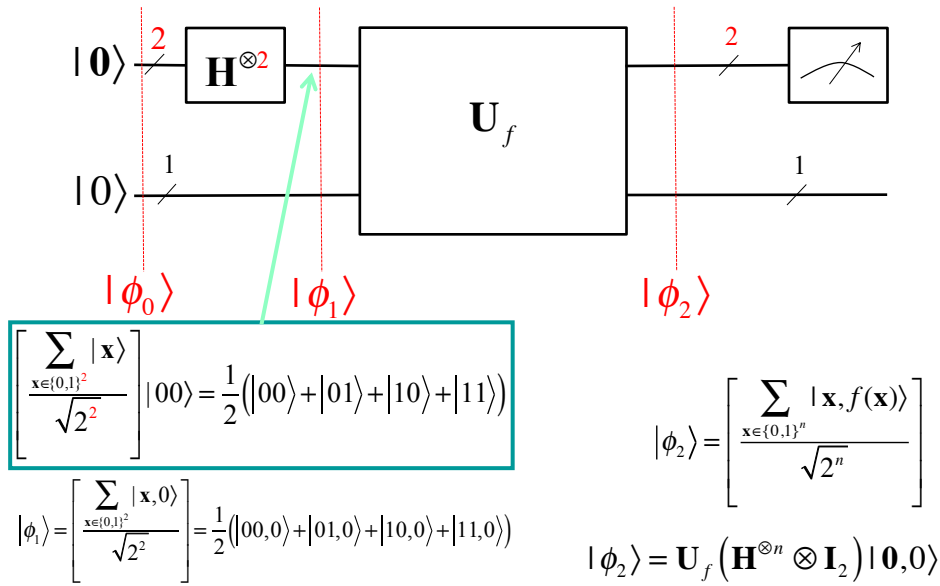
First Attempt to Solve Problem



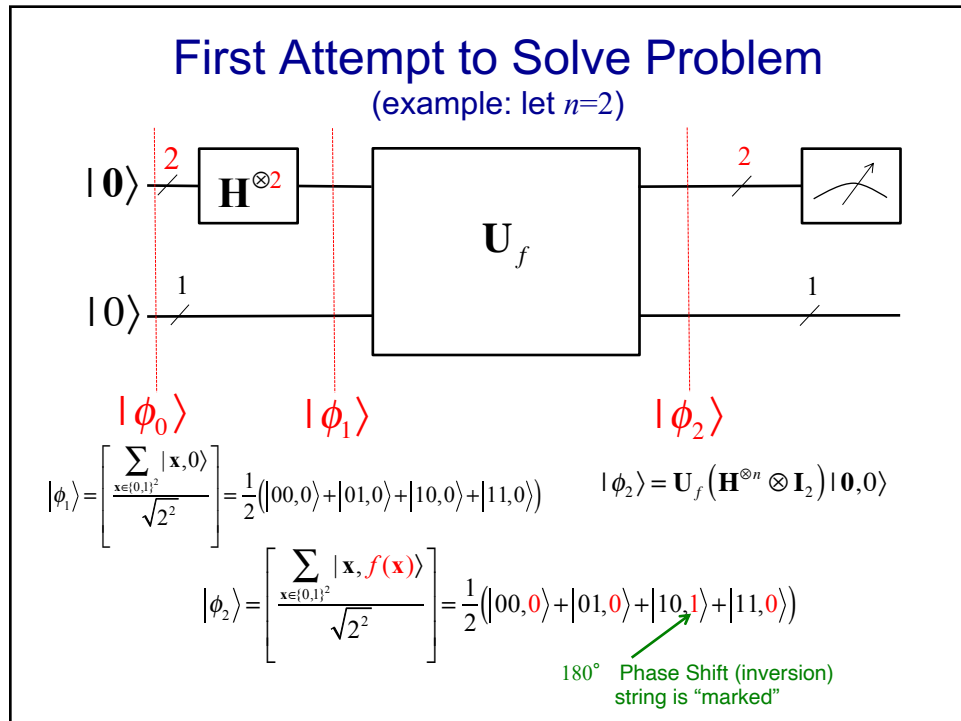
7

First Attempt to Solve Problem

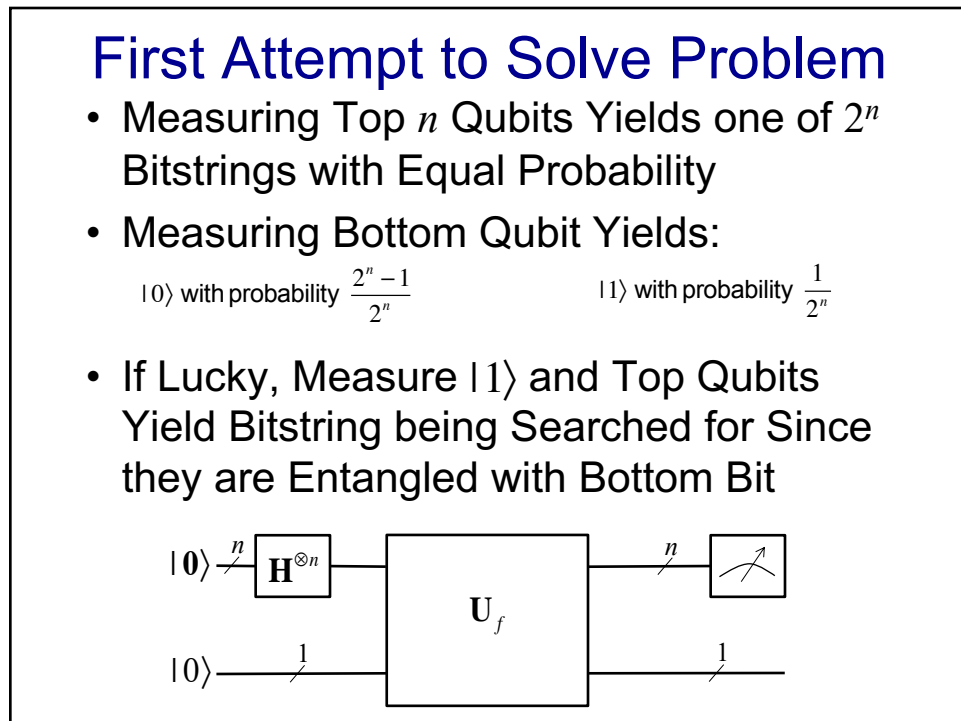
(example: let $n=2$)



8



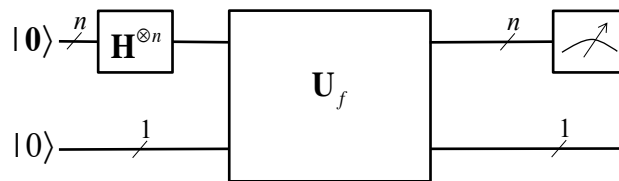
9



10

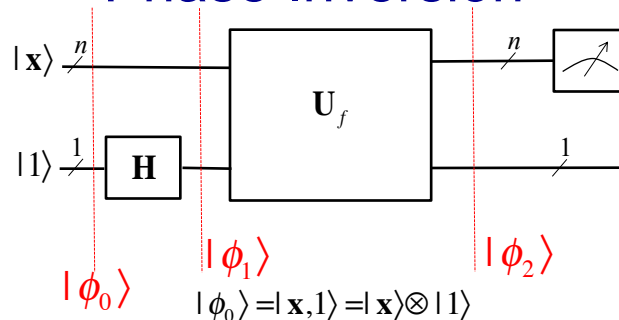
First Attempt to Solve Problem

- Since Chances of Measuring Desired Output are Small, Need Additional Operations
- Use Two New Tricks
 - Phase Inversion
 - Inversion About the Mean (or Average)



11

Phase Inversion



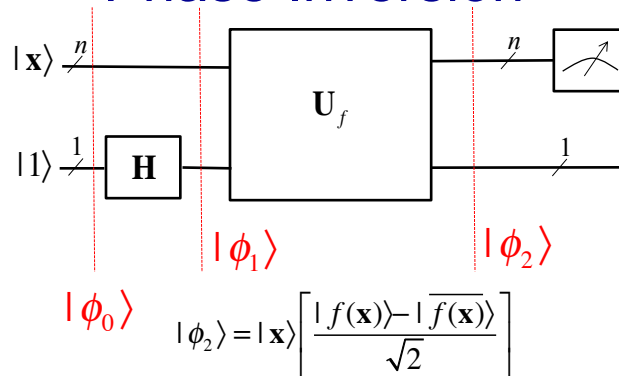
$$|\phi_0\rangle = |x, 1\rangle = |x\rangle \otimes |1\rangle$$

$$|\phi_1\rangle = (\mathbf{I}_n \otimes \mathbf{H}) |x, 1\rangle = |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] = \left[\frac{|x, 0\rangle - |x, 1\rangle}{\sqrt{2}} \right]$$

$$\begin{aligned} |\phi_2\rangle &= (\mathbf{I}_n \otimes \mathbf{H})(U_f) |x, 1\rangle = |x\rangle \left[\frac{|f(x) \oplus 0\rangle - |f(x) \oplus 1\rangle}{\sqrt{2}} \right] \\ &= |x\rangle \left[\frac{|f(x)\rangle - |\overline{f(x)}\rangle}{\sqrt{2}} \right] \end{aligned}$$

12

Phase Inversion



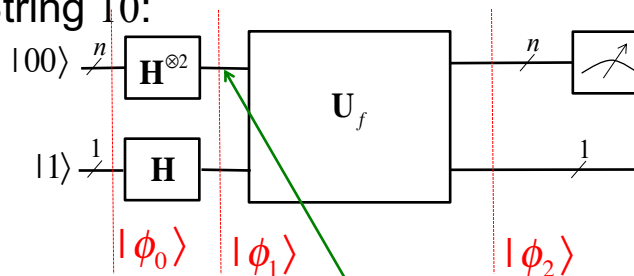
- Since $a-b=(-1)(b-a)$:

$$|\phi_2\rangle = (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle \left[\frac{|10\rangle - |11\rangle}{\sqrt{2}} \right] = \begin{cases} (-1) |\mathbf{x}\rangle \left[\frac{|10\rangle - |11\rangle}{\sqrt{2}} \right], & \text{if } \mathbf{x} = \mathbf{x}_0 \\ (+1) |\mathbf{x}\rangle \left[\frac{|10\rangle - |11\rangle}{\sqrt{2}} \right], & \text{if } \mathbf{x} \neq \mathbf{x}_0 \end{cases}$$

13

Phase Inversion Example

- Assume $n=2$ and Top n Qubits are in Equal State of Superposition and $f(\mathbf{x})$ “Chooses” the String 10:

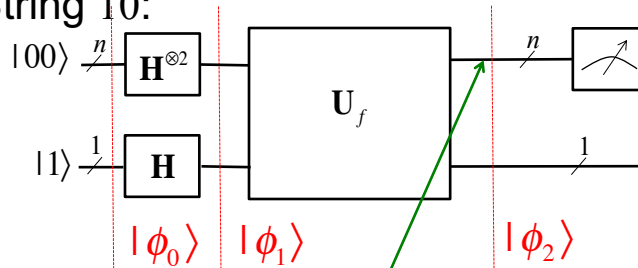


$$(\mathbf{H} \otimes \mathbf{H})|00\rangle = \frac{1}{\sqrt{2^2}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{bmatrix}$$

14

Phase Inversion Example

- Assume $n=2$ and Top n Qubits are in Equal State of Superposition and $f(\mathbf{x})$ “Chooses” the String 10:

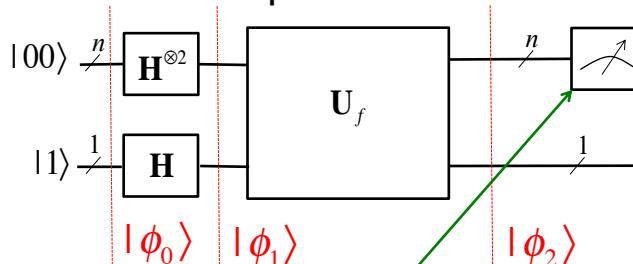


$$(\mathbf{H} \otimes \mathbf{H})(U_f)|00\rangle = \frac{1}{\sqrt{2^2}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix}$$

15

Phase Inversion Example

- Measurement of Top n Qubits:



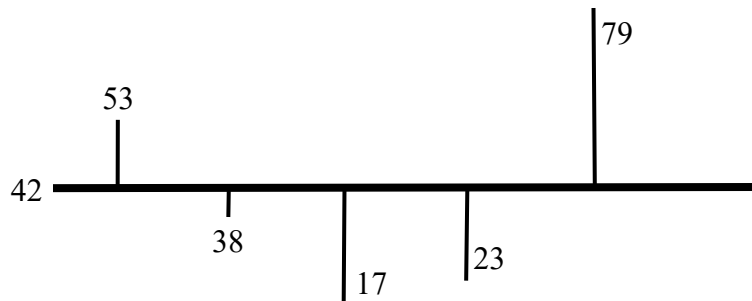
$$\text{Measure} \begin{bmatrix} 1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix}; \quad \text{Prob}(|\mathbf{x}\rangle = 00, 01, 10, 11) = \begin{cases} (1/2)^2, & \mathbf{x} = 00, 01, 11 \\ (-1/2)^2 & \mathbf{x} = 10 \end{cases} = \frac{1}{4}$$

- Need Another “Trick” – Inversion About Mean:

16

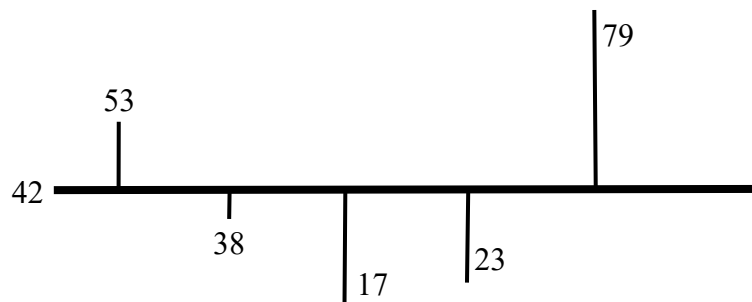
Inversion About the Mean

- Must “Boost” Phase Separation Among Bitstrings
- Use “Inversion About the Mean”
- Consider a Set of Values $\{53, 38, 17, 23, 79\}$:
- Average $\{53, 38, 17, 23, 79\} = 42$



17

Inversion About the Mean



- Desired to Change Sequence so That Each Element Above Average is Same Distance from Average but Below
- Each Element Below Average is Same Distance from Average but Above

18

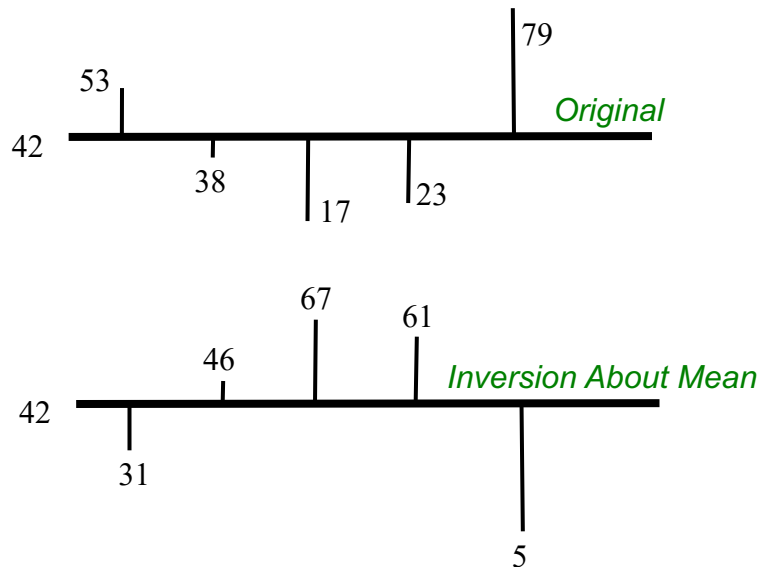
Inversion About the Mean

- To Do This, We INVERT Each Element About Average
 - Move Those Above to Below and vice versa
- EXAMPLE: First Element is 53 and $AVG - 53 = 42 - 53 = -11$ so 11 Units Below (**deviation**)
- Add $AVG = 42$ to Deviation (-11)
- Obtain $AVG + (AVG - 53) = 42 + (42 - 53) = 31$
- Second Element Becomes $42 + (42 - 38) = 46$
- Each Element v Changed to v' :

$$\begin{aligned} v' &= AVG + (AVG - v) \\ &= 2(AVG) - v \end{aligned}$$

19

Inversion About the Mean



20

Inversion About the Mean

- Formulate Inversion About Mean as Matrix Operation
- Consider the Set of Values $\{53, 38, 17, 23, 79\}$:
- We Write the Set as a Column Vector and use an Averaging Matrix, $\mathbf{A}$:

$$\text{average}\{53, 38, 17, 23, 79\} = \begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix} \begin{bmatrix} 53 \\ 38 \\ 17 \\ 23 \\ 79 \end{bmatrix}$$

- Vector of values is $\mathbf{v}^T = [53 \ 38 \ 17 \ 23 \ 79]$
- Vector of averages is $(\mathbf{A}\mathbf{v})^T = [42 \ 42 \ 42 \ 42 \ 42]$

$$\mathbf{v}' = 2(\text{AVG}) - \mathbf{v}$$

$$\mathbf{v}' = 2\mathbf{A}\mathbf{v} - \mathbf{v} = (2\mathbf{A} - \mathbf{I})\mathbf{v}$$

21

Inversion About the Mean

$$\text{average}\{53, 38, 17, 23, 79\} = \begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix} \begin{bmatrix} 53 \\ 38 \\ 17 \\ 23 \\ 79 \end{bmatrix}$$

$$\text{average}\{53, 38, 17, 23, 79\} = \mathbf{A}_5 \mathbf{v} = \frac{53 + 38 + 17 + 23 + 79}{5} = 42$$

$$\mathbf{v}' = -\mathbf{v} + 2\mathbf{A}_n \mathbf{v} = (-\mathbf{I}_n + 2\mathbf{A}_n) \mathbf{v}$$

22

Inversion About the Mean

$$\begin{aligned}
 \mathbf{A}_5 &= \begin{bmatrix} 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{bmatrix} & \mathbf{v} &= \begin{bmatrix} 53 \\ 38 \\ 17 \\ 23 \\ 79 \end{bmatrix} \\
 2\mathbf{A}_5 &= \begin{bmatrix} 2/5 & 2/5 & 2/5 & 2/5 & 2/5 \\ 2/5 & 2/5 & 2/5 & 2/5 & 2/5 \\ 2/5 & 2/5 & 2/5 & 2/5 & 2/5 \\ 2/5 & 2/5 & 2/5 & 2/5 & 2/5 \\ 2/5 & 2/5 & 2/5 & 2/5 & 2/5 \end{bmatrix} & \mathbf{I}_5 &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
 \mathbf{v}' &= -\mathbf{v} + 2\mathbf{A}_n \mathbf{v} = (-\mathbf{I}_n + 2\mathbf{A}_n) \mathbf{v}
 \end{aligned}$$

23

Inversion About the Mean

$$\begin{aligned}
 -\mathbf{I}_5 + 2\mathbf{A}_5 &= \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix} + \begin{bmatrix} 2/5 & 2/5 & 2/5 & 2/5 & 2/5 \\ 2/5 & 2/5 & 2/5 & 2/5 & 2/5 \\ 2/5 & 2/5 & 2/5 & 2/5 & 2/5 \\ 2/5 & 2/5 & 2/5 & 2/5 & 2/5 \\ 2/5 & 2/5 & 2/5 & 2/5 & 2/5 \end{bmatrix} \\
 -\mathbf{I}_5 + 2\mathbf{A}_5 &= \begin{bmatrix} (-1+2/5) & 2/5 & 2/5 & 2/5 & 2/5 \\ 2/5 & (-1+2/5) & 2/5 & 2/5 & 2/5 \\ 2/5 & 2/5 & (-1+2/5) & 2/5 & 2/5 \\ 2/5 & 2/5 & 2/5 & (-1+2/5) & 2/5 \\ 2/5 & 2/5 & 2/5 & 2/5 & (-1+2/5) \end{bmatrix} \\
 \mathbf{v}' &= -\mathbf{v} + 2\mathbf{A}_n \mathbf{v} = (-\mathbf{I}_n + 2\mathbf{A}_n) \mathbf{v}
 \end{aligned}$$

24

Inversion About the Mean

$$\mathbf{v}' = -\mathbf{v} + 2\mathbf{A}_n \mathbf{v} = (-\mathbf{I}_n + 2\mathbf{A}_n) \mathbf{v}$$

$$(-\mathbf{I}_5 + 2\mathbf{A}_5) \mathbf{v} = \begin{bmatrix} (-1+2/5) & 2/5 & 2/5 & 2/5 & 2/5 \\ 2/5 & (-1+2/5) & 2/5 & 2/5 & 2/5 \\ 2/5 & 2/5 & (-1+2/5) & 2/5 & 2/5 \\ 2/5 & 2/5 & 2/5 & (-1+2/5) & 2/5 \\ 2/5 & 2/5 & 2/5 & 2/5 & (-1+2/5) \end{bmatrix} \begin{bmatrix} 53 \\ 38 \\ 17 \\ 23 \\ 79 \end{bmatrix} = \begin{bmatrix} 31 \\ 46 \\ 67 \\ 61 \\ 5 \end{bmatrix}$$

- To Generalize, Consider n Qubits
- Quantum State Vector Contains 2^n Elements
- Form of $\mathbf{A}$ Matrix is:

$$\mathbf{A}_n = \left[\frac{1}{2^n} \right]_{n \times n}$$

25

Inversion About the Mean Property

$$-\mathbf{I}_n + 2\mathbf{A}_n = \begin{bmatrix} (-1+2/2^n) & 2/2^n & \dots & 2/2^n \\ 2/2^n & (-1+2/2^n) & \dots & 2/2^n \\ \vdots & \vdots & \ddots & \vdots \\ 2/2^n & 2/2^n & \dots & (-1+2/2^n) \end{bmatrix}$$

$$\mathbf{B}_n = \mathbf{A}_n^2 = [b_{ij}]_{n \times n}$$

$$b_{ij} = \sum_{k=1}^{2^n} (1/2^n)^2 = (2^n)(1/2^n)^2 = (2^n)(1/2^{2n}) = \frac{2^n}{2^{2n}} = \frac{2^n}{2^{2n}/2^n} = \frac{1}{2^{2n-n}} = \frac{1}{2^n}$$

$$\mathbf{A}_n^2 = \mathbf{A}_n \times \mathbf{A}_n = \mathbf{A}_n$$

$$(-\mathbf{I}_n + 2\mathbf{A}_n)(-\mathbf{I}_n + 2\mathbf{A}_n) = \mathbf{I}_n - 2\mathbf{A}_n - 2\mathbf{A}_n + 4\mathbf{A}_n^2$$

$$(-\mathbf{I}_n + 2\mathbf{A}_n)(-\mathbf{I}_n + 2\mathbf{A}_n) = \mathbf{I}_n - 2\mathbf{A}_n - 2\mathbf{A}_n + 4\mathbf{A}_n$$

$$(-\mathbf{I}_n + 2\mathbf{A}_n)(-\mathbf{I}_n + 2\mathbf{A}_n) = \mathbf{I}_n$$

Unitary Operation!!!!
Realizable as Quantum
Operation!!!

26

Inversion About the Mean Previous 2 Qubit Example

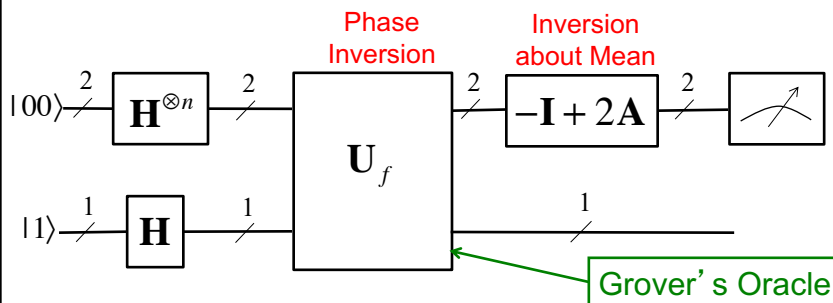
• After Phase Inversion: $\mathbf{v} = \begin{bmatrix} 1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix}$

$$-\mathbf{I}_4 + 2\mathbf{A}_4 = \begin{bmatrix} (-1+2/4) & 2/4 & 2/4 & 2/4 \\ 2/4 & (-1+2/4) & 2/4 & 2/4 \\ 2/4 & 2/4 & (-1+2/4) & 2/4 \\ 2/4 & 2/4 & 2/4 & (-1+2/4) \end{bmatrix}$$

$$(-\mathbf{I}_4 + 2\mathbf{A}_4)\mathbf{v} = \begin{bmatrix} (-1+2/4) & 2/4 & 2/4 & 2/4 \\ 2/4 & (-1+2/4) & 2/4 & 2/4 \\ 2/4 & 2/4 & (-1+2/4) & 2/4 \\ 2/4 & 2/4 & 2/4 & (-1+2/4) \end{bmatrix} \begin{bmatrix} 1/2 \\ 1/2 \\ -1/2 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

27

Putting it All Together



- Classical Computer Search Requires Evaluation of $f(\mathbf{x})$ $4=2^2$ Times
- Quantum Search Requires One Evaluation of Oracle
- For Larger Problems Need to Evaluate Oracle $\sqrt{N} = \sqrt{2^n}$ Times

28

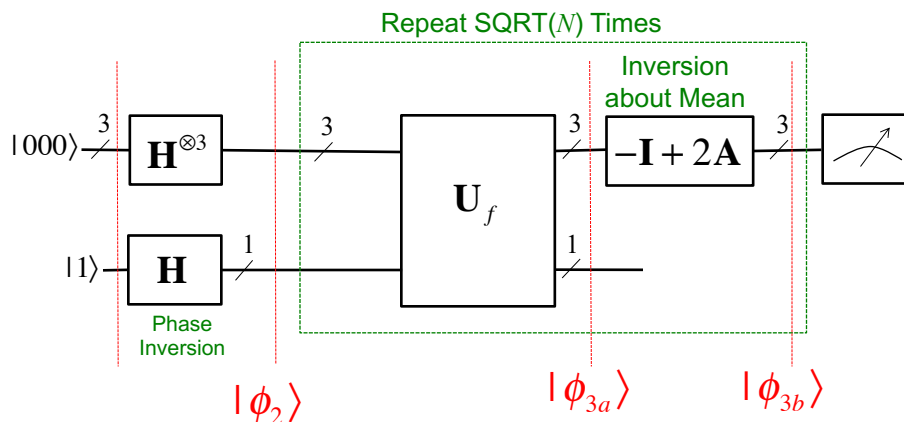
Larger Search Problem

- Consider Search Problem for Bitstring of Length 3
- Oracle Unitary Operation Embeds Boolean Function that Produces a 1 When String 101 is Domain Argument and 0 Otherwise

$$f(\mathbf{x}) = f(x_1x_2x_3) = \begin{cases} 1, & \mathbf{x} = 101 \\ 0, & \mathbf{x} \neq \mathbf{x}_0 \end{cases}$$

29

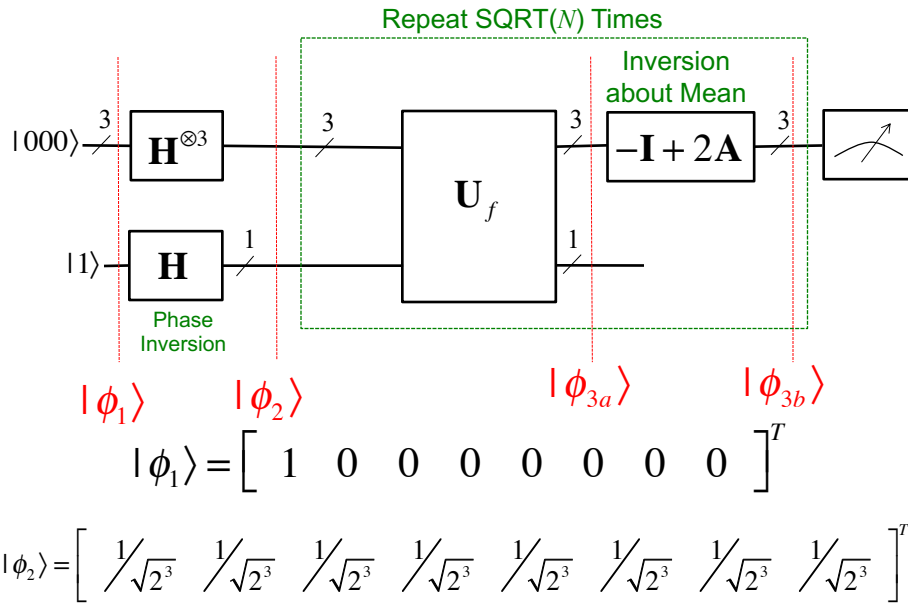
Search Problem for $n=3$



- Following Analysis Considers Top 3 Qubits Only
- Must Cascade Portion in Green Several Times to Enhance Effect of Inversion About Mean
- Example Illustrates this Process

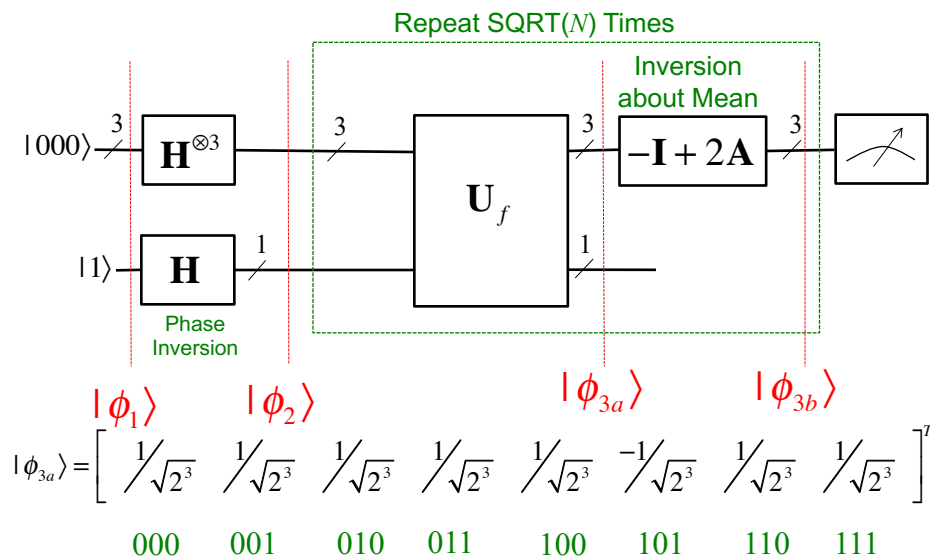
30

Search Problem for $n=3$



31

Search Problem for $n=3$



32

Search Problem for $n=3$

$$|\phi_{3a}\rangle = \begin{bmatrix} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ \frac{1}{\sqrt{2^3}} & \frac{1}{\sqrt{2^3}} & \frac{1}{\sqrt{2^3}} & \frac{1}{\sqrt{2^3}} & \frac{1}{\sqrt{2^3}} & -\frac{1}{\sqrt{2^3}} & \frac{1}{\sqrt{2^3}} & \frac{1}{\sqrt{2^3}} \end{bmatrix}^T$$

- Calculating the Average of These Values:

$$\text{average} = a = \frac{7 \times \frac{1}{\sqrt{8}} - \frac{1}{\sqrt{8}}}{8} = \frac{3}{4\sqrt{8}}$$

- Inversion about Mean for $i = \{000, 001, 010, 011, 100, 110, 111\}$:

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

- Inversion about Mean for 101:

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

33

Search Problem for $n=3$

$$|\phi_{3b}\rangle = \begin{bmatrix} 000 & 001 & 010 & 011 & 100 & 101 & 110 & 111 \\ \frac{1}{2\sqrt{8}} & \frac{1}{2\sqrt{8}} & \frac{1}{2\sqrt{8}} & \frac{1}{2\sqrt{8}} & \frac{1}{2\sqrt{8}} & \frac{5}{2\sqrt{8}} & \frac{1}{2\sqrt{8}} & \frac{1}{2\sqrt{8}} \end{bmatrix}^T$$

- If Measurement Performed Now, Probability of Finding the Search Bitstring is:

$$\text{Prob}[|\phi_{3b}\rangle = |101\rangle] = \left(\frac{5}{2\sqrt{8}}\right)^2 = \frac{25}{32} = 0.78$$

- If Measurement Performed Now, Probability of Finding One of 7 Other Bitstrings is:

$$\text{Prob}[|\phi_{3b}\rangle \neq |101\rangle] = 7 \times \left(\frac{1}{2\sqrt{8}}\right)^2 = \frac{7}{32} = 0.22$$

34

Search Problem for $n=3$

- Desirable to Increase Probability of Measuring the Bitstring we are Searching For
- To Do This, We Phase Invert and Invert About Mean Again
- Implemented By Cascading Green Boxes
- Next Phase Inversion Yields:

$$|\phi_{3c}\rangle = \begin{bmatrix} \text{000} & \text{001} & \text{010} & \text{011} & \text{100} & \text{101} & \text{110} & \text{111} \\ \frac{1}{2\sqrt{8}} & \frac{1}{2\sqrt{8}} & \frac{1}{2\sqrt{8}} & \frac{1}{2\sqrt{8}} & \frac{1}{2\sqrt{8}} & -\frac{5}{2\sqrt{8}} & \frac{1}{2\sqrt{8}} & \frac{1}{2\sqrt{8}} \end{bmatrix}^T$$

$$\text{average} = a = \frac{7 \times \frac{1}{2\sqrt{8}} - \frac{5}{2\sqrt{8}}}{8} = \frac{1}{8\sqrt{8}}$$

35

Search Problem for $n=3$

- Inverting About the Mean Yields:

$$-v_i + 2a = -\frac{1}{2\sqrt{8}} + \left(2 \times \frac{1}{8\sqrt{8}}\right) = -\frac{1}{4\sqrt{8}} \quad -v_i + 2a = \frac{5}{2\sqrt{8}} + \left(2 \times \frac{1}{4\sqrt{8}}\right) = \frac{11}{2\sqrt{8}}$$

$$|\phi_{3d}\rangle = \begin{bmatrix} \text{000} & \text{001} & \text{010} & \text{011} & \text{100} & \text{101} & \text{110} & \text{111} \\ -\frac{1}{4\sqrt{8}} & -\frac{1}{4\sqrt{8}} & -\frac{1}{4\sqrt{8}} & -\frac{1}{4\sqrt{8}} & -\frac{1}{4\sqrt{8}} & \frac{11}{4\sqrt{8}} & -\frac{1}{4\sqrt{8}} & -\frac{1}{4\sqrt{8}} \end{bmatrix}^T$$

- If Measurement Performed Now, Probability of Finding the Search Bitstring is:

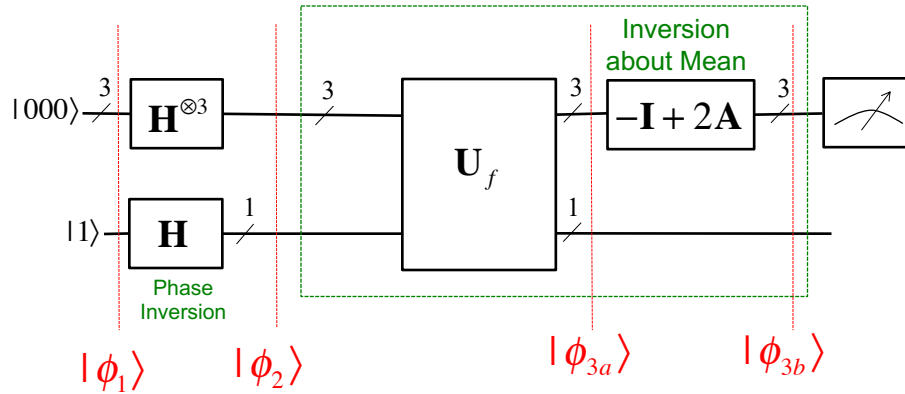
$$\text{Prob}[|\phi_{3d}\rangle = |101\rangle] = \left(\frac{11}{4\sqrt{8}}\right)^2 = \frac{121}{128} = 0.95$$

- If Measurement Performed Now, Probability of Finding One of 7 Other Bitstrings is:

$$\text{Prob}[|\phi_{3d}\rangle \neq |101\rangle] = 7 \times \left(\frac{1}{4\sqrt{8}}\right)^2 = \frac{7}{128} = 0.05$$

36

Search Problem for $n=3$

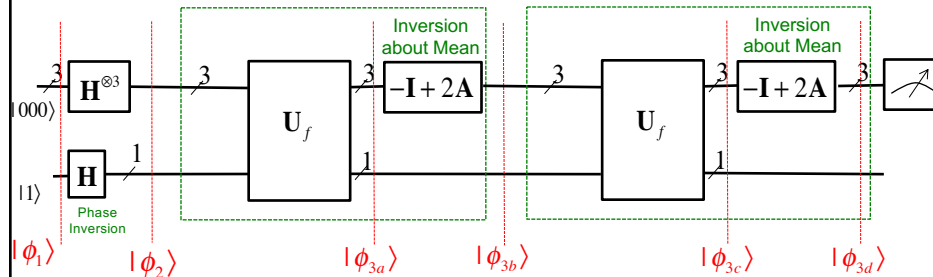


$$\text{Prob}[|\phi_{3b}\rangle = |101\rangle] = \left(\frac{5}{2\sqrt{8}}\right)^2 = \frac{25}{32} = 0.78$$

$$\text{Prob}[|\phi_{3b}\rangle \neq |101\rangle] = 7 \times \left(\frac{1}{2\sqrt{8}}\right)^2 = \frac{7}{32} = 0.22$$

37

Search Problem for $n=3$

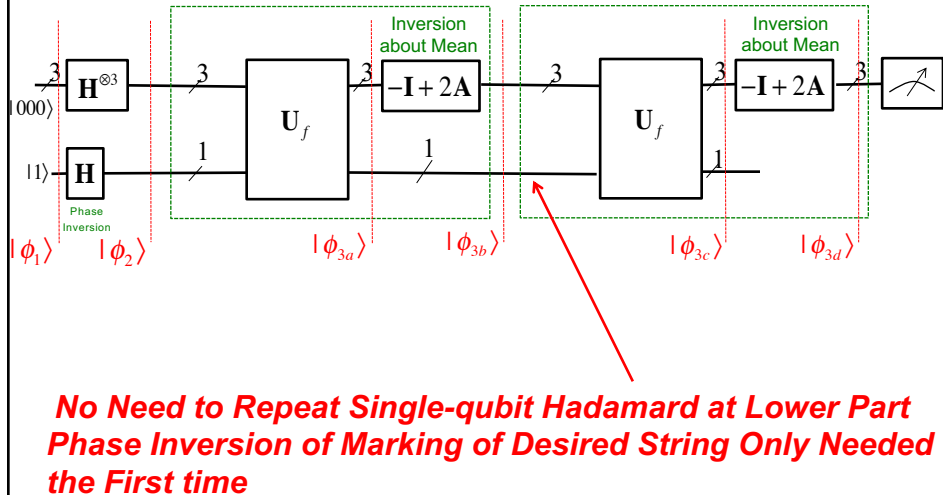


$$\text{Prob}[|\phi_{3d}\rangle = |101\rangle] = \left(\frac{11}{4\sqrt{8}}\right)^2 = \frac{121}{128} = 0.95$$

$$\text{Prob}[|\phi_{3d}\rangle \neq |101\rangle] = 7 \times \left(\frac{1}{4\sqrt{8}}\right)^2 = \frac{7}{128} = 0.05$$

38

Search Problem for $n=3$



39

Grover's Search Algorithm

- Probabilistic Algorithm
- Need to Repeat "Green Box" $\text{SQRT}(N)$ Times
 - $\text{SQRT}(N) = \text{SQRT}(2^n)$ in this Example
- Quadratic Speedup Since Classical Computer Requires N Evaluations and Quantum Computer (implementing Grover's Method) Requires $\text{SQRT}(N)$
- Can Generalize, Search for t Elements Instead of 1 Element Requires $\text{SQRT}(N/t)$ "Green Boxes"

40

Grover's Search Algorithm

- Some Literature Considers “Green Box” to be the “Oracle” – Other Considers the Unitary Operation U_f to be “Oracle”
- Many Problems can be Formulated as Search Problems – Offers Quadratic Speedup
- Must Determine the Oracle – this is the Challenge
- Unlike Other Quantum Algorithms, Grover's Method DOES NOT Provide Exponential Speedup

41

Grover's Search Algorithm

- Importance of Grover's Search is that a QUERY is Accomplished with Quadratic Speedup
- If Oracle Requires Searching through all Strings then no Performance Gain
- Many Modified Versions
- Also, Adaptations to Represent Data in a Quantum Form if it Does NOT Contain All Possible Elements
- Main Contributions: Amplitude Amplification through Inversion about Mean, AND, Phase Inversion

42