CS5100: Quantum Computing

Problem Set - 3

C. Akshay Santoshi
CS21BTECH11012

2.

1. Given $\| |\phi\rangle - |\psi\rangle \| = \sqrt{\sum_i |\alpha_i - \beta_i|^2} = \varepsilon$.

$$\frac{1}{2}\sum_i \left| |\alpha_i|^2 - |\beta_i|^2 \right| = \frac{1}{2}\sum_i \left| (|\alpha_i| - |\beta_i|)(|\alpha_i| + |\beta_i|) \right|$$

$$= \frac{1}{2}\sum_i \left| |\alpha_i| - |\beta_i| \right| (|\alpha_i| + |\beta_i|)$$

$$\leq \frac{1}{2}\sum_i (|\alpha_i - \beta_i|)(|\alpha_i| + |\beta_i|)$$

— Using triangle inequality.

$$\frac{1}{2}\sum_i (|\alpha_i - \beta_i|) \cdot (|\alpha_i| + |\beta_i|)$$

$$\leq \frac{1}{2}\sqrt{\sum_i (|\alpha_i - \beta_i|)^2}\sqrt{\sum_i (|\alpha_i| + |\beta_i|)^2}$$

— Used Cauchy-Schwarz inequality.

Since $2a^2 + 2b^2 \geq (a+b)^2$

$$(a-b)^2 \geq 0$$
$$a^2 + b^2 \geq 2ab$$
$$2a^2 + 2b^2 \geq a^2 + b^2 + 2ab$$
$$\boxed{2a^2 + 2b^2 \geq (a+b)^2}$$

$$\sum_i (|\alpha_i| + |\beta_i|)^2 \leq \sum_i 2|\alpha_i|^2 + \sum_i 2|\beta_i|^2$$

$$\sum_i (|\alpha_i| + |\beta_i|)^2 \leq 2\underbrace{\sum_i |\alpha_i|^2}_{=1} + 2\underbrace{\sum_i |\beta_i|^2}_{1}$$

$$= 4$$

So,

$$\frac{1}{2}\sum_i (|\alpha_i - \beta_i|) \cdot (|\alpha_i| + |\beta_i|) \leq \frac{1}{\cancel{2}}\underbrace{\sqrt{\sum_i |\alpha_i - \beta_i|^2}}_{\text{given as } \varepsilon}\cancel{\sqrt{4}}$$

$$\leq \varepsilon$$

$$\therefore \frac{1}{2}\sum_i \left| |\alpha_i|^2 - |\beta_i|^2 \right| \leq \varepsilon.$$

6

2. $a \in \mathbb{R}^N$ a is r-periodic. $a_l = 1$ if $l \equiv 0 \bmod r$
$a_r = 0$ o/w.

$$F_N = \frac{1}{\sqrt{N}} \begin{array}{c} \\ j \end{array}\!\!\begin{array}{c} k \\ \begin{pmatrix} \cdots\cdots & \omega_N^{jk} & \end{pmatrix} \end{array} \qquad a = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{N-1} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 0 \end{pmatrix} \begin{array}{l} \\ \\ \\ - a_r \\ \\ \\ - a_{2r} \\ \\ \\ - a_{(p-1)r} \\ \\ \\ \end{array}$$

~~$(F_N a)$~~. Let $N = pr$.

$$(F_N a)_j = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \omega_N^{jk} a_k$$

$$= \frac{1}{\sqrt{N}} \sum_{k=0}^{p-1} \omega_N^{jkr}$$

$$= \frac{1}{\sqrt{N}} \sum_{k=0}^{p-1} e^{\frac{2\pi (jkr) i}{N}}$$

$$= \frac{1}{\sqrt{N}} \sum_{k=0}^{p-1} e^{\frac{2\pi (jk) i}{p}} \quad \left(\text{since } \frac{N}{r} = p\right)$$

$$= \frac{1}{\sqrt{N}} \sum_{k=0}^{p-1} (\omega_p^j)^k$$

$$(F_N a)_1 = 1 + \omega_p + \omega_p^2 + \cdots + \omega_p^{(p-1)}$$

$$(F_N a)_2 = 1 + \omega_p^2 + \omega_p^4 + \cdots + \omega_p^{2(p-1)}$$

$$(F_N a)_j = 1 + \omega_p^j + \omega_p^{2j} + \cdots + \omega_p^{(p-1)j}$$

Let $\omega_p = \alpha$.

Case 1: If $\omega_p^j = \alpha^j$ is a primitive p^{th} root of unity, then $1, \alpha^j, (\alpha^j)^2, \ldots (\alpha^j)^{p-1}$ are distinct and the p-roots of unity and

$$\sum_{k=0}^{p-1} \alpha^{kj} = \sum_{k=0}^{p-1} \alpha^k = \sum_{k=0}^{p-1} \omega_p^k = \underline{\underline{0}}.$$

If $j \equiv 0 \bmod p$ then $F_N a_j = 0$.
If not $F_N a_j = \frac{p}{\sqrt{N}}$

Case 2: If $\omega_p^j = 1$, then $\sum_{k=0}^{p-1} \omega_p^{jk} = \sum_{k=0}^{p-1} 1 = p$

This happens when j is a integer multiple of p.

$$(F_N a)_j = \frac{1}{\sqrt{N}} \cdot p = \frac{1}{\sqrt{N}} \cdot \frac{N}{r} = \frac{\sqrt{N}}{r} \text{ if } j = 0 \bmod p = 0 \bmod \left(\frac{N}{r}\right)$$

$$= 0 \text{ if } j \not\equiv 0 \bmod p. \quad \left(p = \frac{N}{r}\right)$$

Non-zero entries are at indexes which are multiple of $\frac{N}{r}$, and their magnitude is $\sqrt{\frac{N}{r}}$ 5

3.(a). $F_N|k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega_N^{jk} |j\rangle$

$$F_N^2|k\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{jk} F_n|j\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} \omega^{jk} \sum_{l=0}^{N-1} \omega^{lj} |l\rangle \times \frac{1}{\sqrt{N}}$$

$$= \frac{1}{N} \sum_{j=0}^{N-1} \sum_{l=0}^{N-1} \omega^{j(l+k)} |l\rangle$$

$$= \frac{1}{N} \sum_{l=0}^{N-1} \left(\sum_{j=0}^{N-1} \omega_N^{j(l+k)} \right) |l\rangle$$

If $\omega_N^{(l+k)}$ is a primitive root of unity, then

$$\sum_{j=0}^{N-1} (\omega_N^{(l+k)})^j = \sum_{j=0}^{N-1} \omega^j = 1 + \omega + .. + \omega^{N-1} = 0$$

If $\omega_N^{(l+k)} = 1$ then $\sum_{j=0}^{N-1} (\omega_N^{(l+k)})^j = \sum_{j=0}^{N-1} 1 = N$.

This happens when $l+k \equiv 0 \bmod N$

$l \equiv (N-k) \bmod N$.

So if $k \neq 0$ $l = N-k$.

if $k=0$ $l=0$

$$F_N^2|k\rangle = \frac{1}{\cancel{N}} \sum_{l=0}^{N-1} (\cancel{N}) |l\rangle = \sum_{l=0}^{N-1} |l\rangle = |N-k\rangle$$ when $k \neq 0$.

So $F_N^2|k\rangle = \begin{cases} |N-k\rangle & k \neq 0 \\ |k\rangle & k = 0 \end{cases}$

$\therefore$ It maps to computational basis states 5

3 (b). ~~When~~ We know $F_N^2|k\rangle \longrightarrow |N-k\rangle$ $k \neq 0$.

$F_N^2|0\rangle \longrightarrow |0\rangle$.

$F_N^2(F_N^2|k\rangle) \longrightarrow F_N^2|N-k\rangle$ $k \neq 0$

$= |N-(N-k)\rangle = |k\rangle$

$F_N^2(F_N^2|0\rangle) \rightarrow F_N^2|0\rangle = |0\rangle$

$\therefore F_N^4|k\rangle \longrightarrow |k\rangle$

~~Therefore~~ $\therefore F_N^4 = I$

If λ is eigen value of F_N with eigenvector v.

$$Iv = F_N^4 v = \lambda^4 v = v$$

$\lambda^4 = 1$ (4th roots of unity)

Eigen values are ± 1, $\pm i$

4

---

4) $|x\rangle \rightarrow |x+y \bmod N\rangle$

$y = y_{n-1} y_{n-2} \cdots y_0 = y_{n-1} \times 2^{n-1} + y_{n-2} \times 2^{n-2} + \ldots + y_0$

Apply QFT.

$$|x\rangle \longrightarrow \frac{1}{\sqrt{N}} \sum_{j \in \{0,1\}^n} e^{\frac{2\pi (jx) i}{N}} |j\rangle$$

Apply single qubit phase shifts

$$\frac{1}{\sqrt{N}} \sum_{j \in \{0,1\}^n} e^{\frac{2\pi (jx+jy) i}{N}} |j\rangle$$

Applying QFT inverse, we will end up in $|x+y \bmod N\rangle$

$$\frac{1}{\sqrt{N}} \sum_{j \in \{0,1\}^n} e^{\frac{2\pi j (x_{n-1}\times 2^{n-1} + x_{n-2}\times 2^{n-2} + \ldots + x_0 + y_{n-1}\times 2^{n-1} + \ldots + y_0) i}{N}} |j\rangle$$

$N = 2^n$

$$e^{2\pi j (x_{n-1}\times 2^{-1} + x_{n-2}\times 2^{-2} + \ldots + x_0 \times 2^{-n} + y_{n-1}\times 2^{-1} + \ldots + y_0 \times 2^{-n}) i} |j_{n-1} j_{n-2} \cdots j_0\rangle$$

$$j = j_{n-1}\times 2^{n-1} + j_{n-2}\times 2^{n-2} + \ldots + j_0 \qquad \text{—— (1)}$$

The first QFT gives phase shift of $e^{\frac{2\pi(jx)i}{N}}$

We should implement gates ~~so~~ that we give a phase shift of $e^{\frac{2\pi(jy)i}{N}}$ ~~so that~~ before doing the QFT^{-1}.

From 2, $e^{\frac{2\pi(jy)i}{N}}$ part would be

$$e^{2\pi j(y_{n-1}\times 2^{-1}+y_{n-2}\times 2^{-2}+\dots+y_0\times 2^{-n})i}\,|j\rangle$$

$$e^{2\pi(j_{n-1}\times 2^{n-1}+j_{n-2}\times 2^{n-2}+\dots+j_0)(y_{n-1}\times 2^{-1}+\dots+y_0\times 2^{-n})i}\,|j_{n-1}j_{n-2}\dots j_0\rangle$$

$$= e^{2\pi(j_{n-1}y_{n-1}\times 2^{n-2}+j_{n-1}y_{n-2}\times 2^{n-3}+\dots+j_{n-1}y_0 2^{-1}+ j_{n-2}y_{n-1}\times 2^{n-3}+j_{n-2}y_{n-2}\times 2^{n-4}+\dots+j_{n-2}y_0\times 2^{-2}+ \;\vdots\; j_0y_{n-1}\times 2^{-1}+j_0y_{n-2}\times 2^{-2}+\dots+j_0\times y_0\times 2^{-n})i}\,|j_{n-1}j_{n-2}\dots j_0\rangle$$

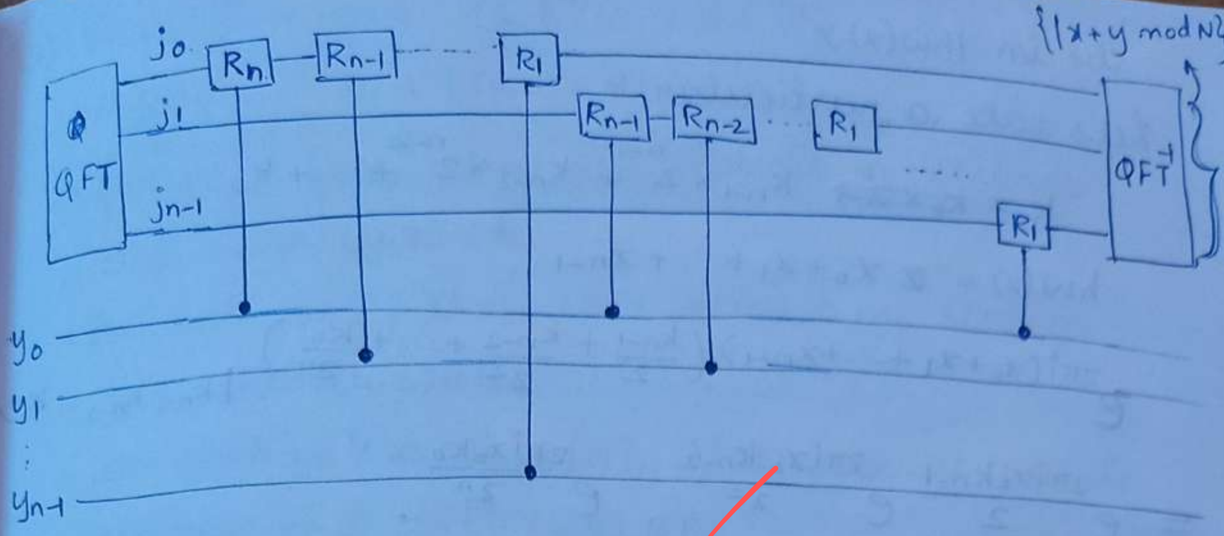
$$= \prod_{l=0}^{n-1}\prod_{l'=0}^{n-1} \cancel{\dots}\, e^{2\pi j_l y_{l'} 2^{l+l'-n} i}\,|j_{n-1}j_{n-2}\dots j_0\rangle$$

$$|j\rangle \longrightarrow \bigotimes_l \left(\prod_{l'=0}^{n-1} e^{2\pi(j_l y_{l'} 2^{l+l'-n})i}\,|j_l\rangle\right)$$

$$\downarrow$$

$$\bigotimes_l |j_l\rangle$$

If you take a particular j_l, you would have phase shifts $R_1, R_2, \dots, R_m$ such that m is largest l' for which $l+l'-n<0$ holds. You can ignore R_{m+1} and so on because we would get $e^{2\pi(j_l y_{l'} 2^{l+l'-n})} = 1$ for higher values.



Values of y :-

If we observe the term $e^{2\pi i (j_l y_{l'} 2^{l+l'-n})}$, the no of gates ($R_s$) we would have would be less if y is large power of 2, because $l+l'-n<0$

If l' is $n-1$, then we would have only one gate R_1.

Would be easier to add for ~~larg~~ higher powers of 2.

Operations required $= 1+2+\ldots+(n-1)+n$

$$= \frac{n(n+1)}{2} \quad \text{(Observe the circuit)}$$

10

$$O(n^2)$$

---

5) $x \in \{0,1\}^n \qquad |x\rangle|0\rangle \longrightarrow |x\rangle|hw(x)\rangle$

$hw(x) \rightarrow$ hamming weight.

Let $x = x_{n-1} \times 2^{n-1} + x_{n-2} \times 2^{n-2} + \ldots\ldots + x_0$

$hw(x) = (x_{n-1} + x_{n-2} + \ldots + x_0)$

$$0^n \xrightarrow{QFT} \frac{1}{\sqrt{2^n}} \sum_k |k\rangle$$

We want $\frac{1}{\sqrt{2^n}} \sum_k |k\rangle \longrightarrow \frac{1}{\sqrt{2^n}} \sum_k e^{\frac{2\pi i (hw(x))k}{2^n}} |k\rangle$.

so that after we apply QFT^{-1} we would be in $|hw(x)\rangle$

Let's take a particular k

$$k = \cancel{k_0 \times 2^0 +} k_{n-1} \times 2^{n-1} + k_{n-2} \times 2^{n-2} + \dots + k_0.$$

$$hw(x) = x_0 + x_1 + \dots + x_{n-1}$$

$$e^{2\pi i (x_0 + x_1 + \dots + x_{n-1})\left(\frac{k_{n-1}}{2} + \frac{k_{n-2}}{2^2} + \dots + \frac{k_0}{2^n}\right)} |k_{n-1} k_{n-2} \dots k_0\rangle$$

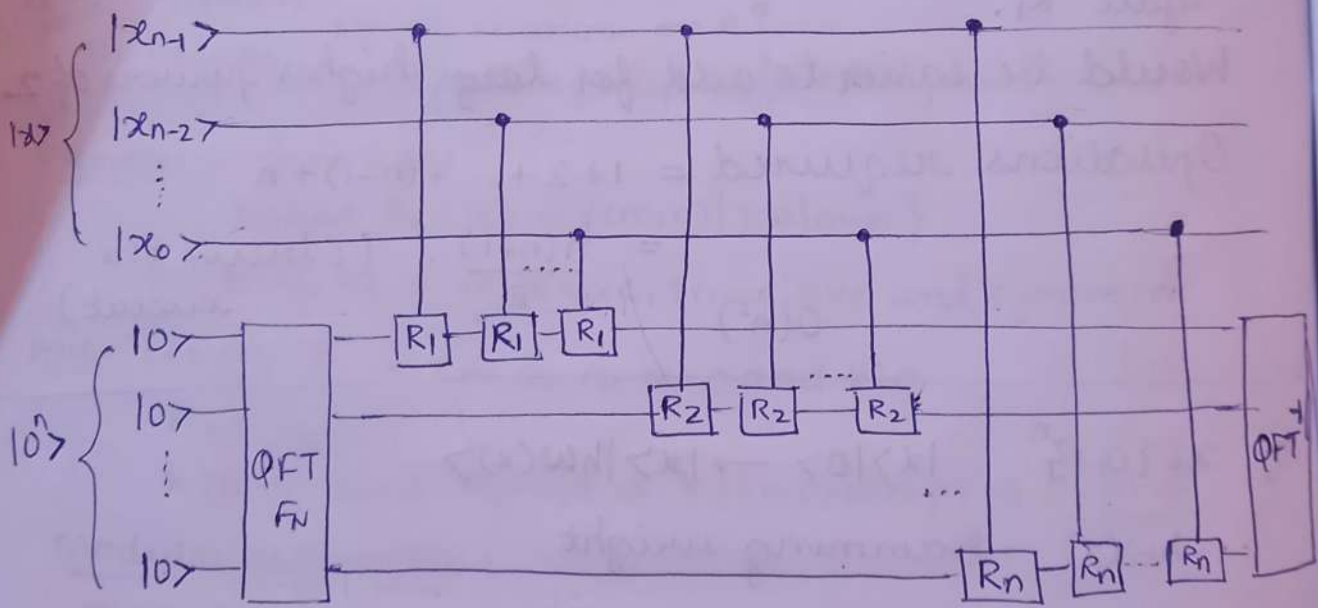
$$= e^{\frac{2\pi i x_0 k_{n-1}}{2}} e^{\frac{2\pi i x_0 k_{n-2}}{2^2}} \dots e^{\frac{2\pi i x_0 k_0}{2^n}} \cdot$$

$$e^{\frac{2\pi i x_1 k_{n-1}}{2}} e^{\frac{2\pi i x_1 k_{n-2}}{2^2}} \dots e^{\frac{2\pi i x_{n-1} k_0}{2^n}} \cdot$$

$$\vdots$$

$$e^{\frac{2\pi i x_{n-1} k_{n-1}}{2}} e^{\frac{2\pi i x_{n-1} k_{n-2}}{2^2}} \dots e^{\frac{2\pi i x_{n-1} k_0}{2^n}} \cdot |k_{n-1} k_{n-2} \dots k_0\rangle$$

$$= \bigotimes_{l} \left(\prod_{j=0}^{n-1} e^{\frac{2\pi i x_j k_l}{2^{n-l}}} |k_l\rangle \right)$$



The final state would be $|x\rangle|hw(x)\rangle$

Size of circuit $= \underbrace{n + n + \dots + n}_{n}. = n^2$

$O(n^2)$

0

13) $f: \{0,1\}^N \to \{0,1\}$ $Y \subseteq f^{-1}(1)$ $Z \subseteq f^{-1}(0)$

$R \subseteq Y \times Z$ $R_j = \{(y,z) \in R \mid y_j \neq z_j\}$.

- for each $y \in Y$, there are atleast m_1 strings $z \in Z$ with $(y,z) \in R$.
- for each $z \in Z$, there are atleast m_0 strings $y \in Y$ with $(y,z) \in R$.
- for each $y \in Y$ and $j \in [N]$, there are at most l_1 strings $z \in Z$ with $(y,z) \in R_j$.
- for each $z \in Z$ and $j \in [N]$, there are at most l_0 strings $y \in Y$ with $(y,z) \in R_j$.

Following notations of class:

$$\text{Progress}_t = \sum_{(y,z) \in R} |\langle \psi_y^t | \psi_z^t \rangle|$$

$$\text{Progress}_0 = \sum_{(y,z) \in R} |\langle \psi_y^0 | \psi_z^0 \rangle| = \sum_{(y,z) \in R} 1 = |R| \quad \cancel{|R| \leq R}$$

~~since $R \subseteq Y \times Z$~~

Bound $\text{Progress}_t - \text{Progress}_{t+1}$.

$|\psi_y^t\rangle = \sum_i \alpha_i |i\rangle \otimes |\phi_i\rangle$ where $|\phi_i\rangle$ is a unit vector and $\sum_{j=1}^N |\alpha_i|^2 = 1$

$|\psi_z^t\rangle = \sum_i \beta_i |i\rangle \otimes |\chi_i\rangle$ where $|\chi_i\rangle$ is a unit vector and $\sum_{i=1}^N |\beta_i|^2 = 1$

$$\langle \psi_y^t | \psi_z^t \rangle = \bar{\alpha}_1 \beta_1 \langle \phi_1 | \chi_1 \rangle + \bar{\alpha}_2 \beta_2 \langle \phi_2 | \chi_2 \rangle + \dots + \bar{\alpha}_N \beta_N \langle \phi_N | \chi_N \rangle$$

$$\langle \psi_y^{t+1} | \psi_z^{t+1} \rangle = \sum_{i: y_i = z_i} \bar{\alpha}_i \beta_i \langle \phi_i | \chi_i \rangle - \sum_{i: y_i \neq z_i} \bar{\alpha}_i \beta_i \langle \phi_i | \chi_i \rangle.$$

$\text{Progress}_t - \text{Progress}_{t+1}$

$$= \sum_{(y,z)\in R} |\langle \psi_y^t | \psi_z^t \rangle| - \sum_{(y,z)\in R} |\langle \psi_y^{t+1} | \psi_z^{t+1} \rangle|$$

$$\le \sum_{(y,z)\in R} |\langle \psi_y^t | \psi_z^t \rangle - \langle \psi_y^{t+1} | \psi_z^{t+1} \rangle|$$

(Triangle Inequality

$$\langle \psi_y^t | \psi_z^t \rangle - \langle \psi_y^{t+1} | \psi_z^{t+1} \rangle = 2 \sum_{i: y_i \neq z_i} \overline{\alpha_i} \beta_i \langle \phi_i | \chi_i \rangle$$

$\text{Progress}_t - \text{Progress}_{t+1}$

$$\le \sum_{(y,z)\in R} 2 \left| \sum_{i: y_i \neq z_i} \overline{\alpha_i} \beta_i \langle \phi_i | \chi_i \rangle \right|$$

$$\le \sum_{(y,z)\in R} 2 . \sum_{i: y_i \neq z_i} |\alpha_{i(y,z)}| |\beta_{i(y,z)}|$$

$$= \sum_{i=1}^{N} \sum_{(y,z)\in R_i} 2 . |\alpha_{i(y,z)}| |\beta_{i(y,z)}|$$

$$\le \sum_{i=1}^{N} \left(\sum_{(y,z)\in R_i} \sqrt{\frac{l_0 m_1}{l_1 m_0}} |\alpha_{i(y,z)}|^2 + \sum_{(y,z)\in R_i} \sqrt{\frac{l_1 m_0}{l_0 m_1}} |\beta_{i(y,z)}|^2 \right)$$

Used $2ab \le h . a^2 + \left(\frac{1}{h}\right) . b^2$ $h = \sqrt{\frac{l_0 m_1}{l_1 m_0}}$

Given that for a particular y and $i \in [N]$, there are atmost l_1 strings $z \in Z$ s.t $(y,z) \in R_i$

Similarly for particular z and $i \in [N]$, there are atmost l_0 strings $y \in Y$ s.t $(y,z) \in R_i$

$\text{Progress}_t - \text{Progress}_{t+1} \le$

$$\sum_{j=1}^{N} \left(\sum_{y\in R} l_1 \sqrt{\frac{l_0 m_1}{l_1 m_0}} |\alpha_{i(y,z)}|^2 + \sum_{z\in R} l_0 \sqrt{\frac{l_1 m_0}{l_0 m_1}} |\beta_{i(y,z)}|^2 \right.$$

i

We know $|R| \geq m_1|Y|$ and $|R| \geq m_0|Z|$

$\text{Progress}_t - \text{Progress}_{t+1} \leq$

$$\sum_{y\in R} l_1 \sqrt{\frac{l_0 m_1}{l_1 m_0}} \underbrace{\sum_{j=1}^{N} |\alpha_{i(y,z)}|^2}_{\leq 1} + \sum_{z\in R} l_0 \sqrt{\frac{l_1 m_0}{l_0 m_1}} \underbrace{\sum_{j=1}^{N} |\beta_{i(y,z)}|^2}_{\leq 1}$$

$$\leq \sum_{y\in R} l_1 \sqrt{\frac{l_0 m_1}{l_1 m_0}} + \sum_{z\in R} l_0 \sqrt{\frac{l_1 m_0}{l_0 m_1}}$$

$$\leq l_1 \sqrt{\frac{l_0 m_1}{l_1 m_0}} |Y| + l_0 \sqrt{\frac{l_1 m_0}{l_0 m_1}} |Z|$$

$$\leq l_1 \sqrt{\frac{l_0 m_1}{l_1 m_0}} \frac{|R|}{m_1} + l_0 \sqrt{\frac{l_1 m_0}{l_0 m_1}} \frac{|R|}{m_0}$$

$$= \frac{2|R|\sqrt{l_0 l_1}}{\sqrt{m_0 m_1}}$$

~~$Q(f) \geq \Omega$~~ $\text{Progress}_0 = |R|$ $\text{Progress}_T \leq 0.99|R|$

$$T \geq \Omega\left(\sqrt{\frac{m_0 m_1}{l_0 l_1}}\right)$$

$$\therefore Q(f) \geq \Omega\left(\sqrt{\frac{m_1 m_0}{l_0 l_1}}\right)$$

If we measure $|\psi_y^T\rangle$ we output 1 with probability atleast 2/3.

If we measure $|\psi_z^T\rangle$ we output 0 with probability atleast 2/3.

10

14). $f^{-1}(1) \supseteq Y$ = { set of strings with exactly k 1's }.

$f^{-1}(0) \supseteq Z$ = { ~~string of all 0~~ 0^n string will all bits as 0 }

$R = Y \times Z$

$R_j = \{(y,z) \in R \mid y_j \neq z_j\}$

$m_1 = 1$ (for each $y \in R$, we would have only one string z, which is all 0's)

$m_0 = {}^NC_k$ (no of possible y's for particular z should have 1's at exactly k positions)

$l_1 = 1$ (Only the all 0's present in Z)

$l_0 = {}^{N-1}C_{k-1}$ (For a particular i, $z_i = 0$, $y_i = 1$, so ~~atmost~~ ~~there~~ we can have $(k-1)$ 1's in the remaining $(N-1)$ positions since hamming weight is k)

$$\sqrt{\frac{m_0 m_1}{l_0 l_1}} = \sqrt{\frac{({}^NC_k)}{({}^{N-1}C_{k-1})}} = \sqrt{\frac{N!}{k!(N-k)!} \times \frac{(N-k)!(k-1)!}{(N-1)!}}$$

$$= \sqrt{\frac{N}{k}}$$

Query lower bound - $\Omega\left(\sqrt{\frac{N}{k}}\right)$

4

15) $N = k^2$. given k is odd. Let $k = 2p-1$.

$f : \{0,1\}^N \to \{0,1\}$

$f(x) = \mathrm{Maj}_k(\mathrm{OR}_k(x^{(1)}), \ldots, \mathrm{OR}_k(x^{(k)}))$

$x = x^{(1)} \ldots x^{(k)}$ $\quad x^{(i)} \in \{0,1\}^k$.

$f^{-1}(1) \supseteq Y = \{$ Exactly p of the k ORs in the MAJ evaluate to 1 and if an $\mathrm{OR}(x^{(i)}) = 1$, then there is 1 at exactly one position and zero in the remaining positions of $x^{(i)}\}$

$f^{-1}(0) \supseteq Z = \{$ Exactly $p-1$ of the k ORs in the MAJ evaluate to 1 and if an $\mathrm{OR}(x^{(i)}) = 1$, then there is 1 at exactly one position and zero in the remainining positions of $x^{(i)}\}$.

$(y,z) \in R$ if it satisfies:
$y \in Y$, $z \in Z$.
y and z differ at only position.

$R_j = \{(y,z) \in R \mid y_j \neq z_j\}$.

$m_1 = ({}^{p}C_1)$ $\{$ since $(y,z) \in R$ only if y and z differ at one place, for a particular y, there can choose one $x^{(i)}$ of the p OR's which evaluate to 1 in y and make that 0 for z$\}$.

$m_0 = ({}^{p}C_1)({}^{k}C_1)$ $\{$ For particular z, we need to have same portion as z for the places in y, ~~and in the remaining~~

~~which ever~~ for which $OR(x^{(i)})=1$ in z, now ~~at remai~~ in the remaining p ORs we can select one OR block, and in that OR block choose one of the k places ~~and~~ to ~~keep~~ be kept as 1 }

$l_0 = 1$ (~~if~~ ~~y_i~~ Case $y_i=0$ and $z_i=1$ is not possible from how we have defined $(y,z)\in R$.
The blocks at which OR becomes 1 in z would also become 1 in y and there would be one additional OR block in y which evaluates to 1.
Case $y_i=1$ and $z_i=0$ would be a single string which has all places $j\neq i$ same as y)

$l_1 = 1$ (Case $z_i=1$ and $y_i=0$ is not possible.
Case $z_i=0$ and $y_i=1$ would be one string in y with same reasons as above)

$$\sqrt{\frac{m_0 m_1}{l_0 l_1}} = \sqrt{\frac{({}^{p}C_1)\times({}^{p-1}C_1)({}^{k}C_1)}{1}} = \sqrt{p(p-1)k}$$

$$= \sqrt{\left(\frac{k+1}{2}\right)\left(\frac{k+1}{2}\right)k} = \frac{1}{2}\sqrt{k^3+2k^2+k}$$

$$\approx \Omega(k^{3/2})$$

$$= \Omega(N^{3/4})$$

6

10) $x = x_0 x_1$ $\quad O_{x,\pm} : |i\rangle \mapsto (-1)^{x_i} |i\rangle$

$H O_{x,\pm} H |0\rangle$

$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$

$O_{x,\pm} H|0\rangle = \frac{1}{\sqrt{2}}\left((-1)^{x_0}|0\rangle + (-1)^{x_1}|1\rangle\right)$

$H O_{x,\pm} H|0\rangle = \frac{(-1)^{x_0}}{\sqrt{2}}\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) + \frac{(-1)^{x_1}}{\sqrt{2}}\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right)$

$= \left(\frac{(-1)^{x_0} + (-1)^{x_1}}{2}\right)|0\rangle + \left(\frac{(-1)^{x_0} - (-1)^{x_1}}{2}\right)|1\rangle$

$(-1)^{x_i} = \begin{cases} -1 & \text{if } x_i = 1 \\ 1 & \text{if } x_i = 0 \end{cases}$

$(-1)^{x_i}$ can be written as $(1 - 2x_i)$

Probability that it outputs 1 is

$\left(\frac{(-1)^{x_0} - (-1)^{x_1}}{2}\right)^2 = \left(\frac{(1-2x_0) - (1-2x_1)}{2}\right)^2 = (x_1 - x_0)^2$

$= x_1^2 + x_0^2 - 2x_1 x_0$

if $x_1 = x_0$ $\quad Pr(\text{output} = 1) = 0$

if $x_1 \neq x_0$ $\quad Pr(\text{output} = 1) = 1$

This is like parity function on 2-bit input.

5

---

11) (a) Previous question algo computed parity on 2 bit input.

Now, we have input $x = x_0 x_1 \dots x_N$ where N is given as even.

$x = \underbrace{(x_0 x_1)(x_2 x_3) \dots}_{N/2 \text{ pairs}} (x_{N-1} x_N)$

Divide input into $N/2$ pairs as shown. Now apply the 1-query algo $H O_{x,\pm} H|0\rangle$ on the

2-bit pair for each of the $N/2$ pairs.
Now, we can ~~xor~~ XOR the $N/2$ results to get the parity of n-bits. We don't require any additional quantum queries.
Parity success probability is 1 and we have used only $N/2$ queries.

5

11)(b). On every input, consider algo's that have error probability $\leq 1/3$.
Parity just depends on no. of 1's in input, whether even no or odd no.
So we can define function which computes parity as

$$g : \{0,1,\ldots,n\} \rightarrow \mathbb{R}$$

$$g(x) = \begin{cases} 0, & \text{if } x \text{ is even} \\ 1, & \text{if } x \text{ is odd.} \end{cases}$$

From question we have

$$|g(x) - f(x)| \leq 1/3 \quad \text{for all } x \in \{0,1\}^n$$

For this, we can have

$$|G(y) - F(y)| = |\underset{\substack{\text{Hamming} \\ \text{weight} \\ \text{of } x = y}}{\mathbb{E}} (g(x) - f(x))| \leq 1/3$$

in that case you will have one polynomial for each hamming wt

X

Thus, we have $G(y)$ which is multilinear poly. that 1/3-approximates parity function. $F(y)$

If y is even $(|G(y)| \leq 1/3) \Rightarrow -\frac{1}{3} \leq G(y) \leq 1/3$

If y is odd $(|G(y)-1| \leq 1/3) \Rightarrow \frac{2}{3} \leq G(y) \leq \frac{4}{3}$

Mean value theorem states that for a closed differentiable interval $[a,b]$, there exists c in (a,b) s.t $f'(c) = \frac{f(b)-f(a)}{b-a}$.

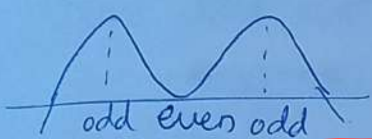
~~So~~ Let a = odd no, b = even

$\cancel{g'(c) = \frac{g(\text{even}) - g(\text{odd})}{1}} \leq$

$G'(c) = \frac{G(\text{even}) - G(\text{odd})}{1} \leq \frac{1}{3} - \frac{2}{3} = \frac{-1}{3}$.

Let a = even, b = odd

$G'(c) = \frac{G(\text{odd}) - G(\text{even})}{1} \geq \frac{2}{3} - \frac{1}{3} = \frac{1}{3}$.



→ possibility of $G(y)$.

?

So ~~the~~ $G'(y)$ can have atleast $(n-1)$ roots,

so the approximating poly. $G(y)$ can have degree atleast n.

This is true ~~for univariate~~. What you have defined is multi variate.

We know that, for a ~~t-q~~ T-query algo, for ~~a prob~~ input $x \in \{0,1\}^n$, the accepting probability is a polynomial in $x_1, x_2, \ldots, x_n$ of degree at most $2T$.

$\therefore 2T \geq n \Rightarrow t \geq n/2$.

$\therefore$ Optimal bound even for algos ~~with~~ which have error probability $\leq 1/3$ on every input.

6

12) Theorem from class:

Let A be a quantum query algo making T queries. Then amplitudes of the final state are multilinear polynomials each with degree atmost T. over complex

$$\sum_{x \in \{0,1\}^m} \alpha(x) |x\rangle.$$

Given that we have a T-query algo that computes N-bit OR function with success probability 1.

From theorem, the final state ~~is~~ can be written as $\sum_{\cancel{x \in \{0,1\}^m}\; k \in \{0,1\}^m} \alpha_k(X) |k\rangle$

Let $S = \{$ set of all basis states whose output came out to be 1 $\}$.

If $k \in S$, then $\alpha_K(X) = 0$ whenever the input is not 0^n because if not, we would ~~get~~ be ~~some other state~~ violating the statement that the ~~function~~ algo has success probability equal to 1.

For input 0^n, since probability of getting output as 0 is 1 positive, there exists a ~~sta~~ state $l \in S$ s.t. $P_l(0^n) \neq 0$ X How did it become $\neq 0$?

6

Now, consider the polynomial $p(x)$ as the $Re\left(1 - \frac{P_l(x)}{P_l(0^n)}\right)$ to represent OR function.

This has degree atmost T. We know that

OR function should have degree atleast N since $OR(x) = 1-(1-x_1)(1-x_2)\cdots(1-x_n)$

$\therefore \underline{\underline{T \geq N}}$.

---

9) $G=(V,E)$ $M :=$ adjacency matrix of G.

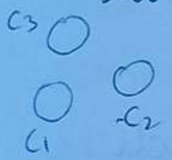
$$O_M : |i,j,b\rangle \longmapsto |i,j,b \oplus M_{ij}\rangle$$

(a). Given G is connected.

G_A is subgraph with (V,A) $A \subseteq E$.

G_A has c connected components.

Edge $(i,j) \in E$ is "good" if it connects any two the components.



~~If~~ These individual components can be assumed to be isolated vertices of a graph. So for them to be connected, it should have atleast $(c-1)$ edges (eg: Path graph).

$\therefore$ No. of good edges are atleast $c-1$.

We can have a query $M'_{ij,\pm}$, ~~same~~ similar to $O_{x,\pm}$ s.t. whenever we provide i and j, it ~~the~~ maps $|i,j\rangle$ to $(-1)^{M'_{ij}}|i,j\rangle$ where $M'_{ij}=1$ if ij is a good edge and $M'_{ij}=0$, if not.

So, we can ~~do~~ use Grover's algo., which uses $M'_{ij,\pm}$ instead of $O_{x,\pm}$ whenever query is asked.

So, expected time was $O\left(\sqrt{\frac{N}{t}}\right)$. In our case,

$N =$ no. of edges in graph $= \frac{N(N-1)}{2}$.

t is atleast $c-1 \Rightarrow t \geq c-1$.

Since graph is given to be connected, this algo will find a good edge and terminate with an expected no of $O(\cancel{\frac{n}{\sqrt{}}})$

$$O\left(\sqrt{\frac{N(N-1)}{2(c-1)}}\right) \approx O\left(\frac{N}{\sqrt{c-1}}\right) \text{ queries.}$$

10

9(b) Quantum algo to decide whether G is connected or not:

Initially, we do not know any edge, so we just have n isolated vertices, n components.

We can use the previous ~~quantum~~ grover search ~~alt~~ modified algo to find a good edge. This takes $O\left(\frac{N}{\sqrt{N-1}}\right)$ queries.

Now to our modified subgraph which contains N-1 connected components, we can ~~call~~ use the previous algo again.

~~We continue to do~~. This takes $O\left(\frac{N}{\sqrt{N-2}}\right)$ queries

We can continue to do this until the no of connected components is reduced to 1 or the algo hasn't terminated yet after $\cancel{O\left(\frac{N}{\sqrt{c-1}}\right)}$ after a ~~lot~~ no. of queries which we give a bound below:

If graph is connected, algo terminates after an expected no of

$$O\left(\frac{N}{\sqrt{N-1}}\right) + O\left(\frac{N}{\sqrt{N-2}}\right) + \dots + O\left(\frac{N}{1}\right) \text{ queries}$$

$$\leq O\left(\sum_{i=1}^{N-1} \frac{N}{\sqrt{i}}\right) \leq O\left(\int_{x=1}^{N-1} \frac{N}{\sqrt{x}}\, dx\right)$$

$$= O\left(N\left[\frac{x^{-1/2+1}}{1/2}\right]_1^{N-1}\right) = O\left(N\left(2(N-1)^{1/2}-2\right)\right)$$

$$\approx O(N^{3/2}) \text{ queries.}$$

Since given success probability is $\geq 2/3$, if algo hasn't terminated after ~~30~~ ~~3 times~~ ~~the ex~~ ~~3 ti~~ $3kN^{3/2}$ queries where $kN^{3/2}$ is the expected no of queries before getting terminated for a connected graph, we can ~~out~~ decide that the graph is not connected ~~which has the given success probability~~.

If it has terminated within the given no, then we can decide that the graph is connected. ~~If~~ It has a success probability of $\geq 2/3$.

10

---

8) Let us have a set S which initially contains first s ~~elements~~ x_i values we have queried for. We will lower bound this s later.

So initially in these s elements, if we find a collision then we can output the pair and terminate with success probability $\geq 2/3$.

If we haven't found a collision, then we ~~ca~~ know that ~~apo~~ there are s other ~~elements~~ ~~out~~ ~~to~~ x_i's outside of S which will have a collision with ~~an elem~~ an element in S. This is because, we are promised that we have an (i,j) pair for every i such that $x_i = x_j$ and $i \neq j$

So we can use Grover search ~~subro~~ subroutine to find an element outside S that gives us a collision. This takes an expected no of $O\left(\sqrt{\frac{n}{t}}\right)$ queries, where for our question $n = N$ ~~(or $N-r$, since we have already queried r)~~ and $t \simeq S$

Total no of queries $= O\left(\underset{\downarrow}{S} + \sqrt{\frac{N}{S}}\right)$

This is for initial S queries

Lower bound: Optimal when $S + \sqrt{\frac{N}{S}}$ attains its lowest. So

$$\frac{d}{ds}\left(s + \sqrt{\frac{N}{s}}\right) = 0.$$

$$1 + \sqrt{N}\left(\frac{-1}{2}\right) s^{-3/2} = 0 \Rightarrow s^{3/2} = \frac{N^{1/2}}{2}$$

$$\Rightarrow s = \frac{N^{1/3}}{2^{2/3}}$$

$$\text{Queries} = O\left(\frac{N^{1/3}}{2^{2/3}} + \sqrt{\frac{N}{N^{1/3}/2^{2/3}}}\right)$$

$$\approx O(N^{1/3} + N^{1/3}) \approx O(N^{1/3})$$

This quantum algo finds collision with probability $\geq 2/3$ using $O(N^{1/3})$ queries.

10

6)(b) $\mathcal{A} = H^{\otimes n} \otimes U_\gamma$

$\mathcal{A}|0^{n+1}\rangle = H^{\otimes n}|0^n\rangle \otimes \begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix}\begin{pmatrix} 1 \\ 0 \end{pmatrix}$

$= \frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} |k\rangle \otimes (\cos\gamma|0\rangle + \sin\gamma|1\rangle)$

$= \frac{1}{\sqrt{2^n}} \sum_{k\in\{0,1\}^n} (\cos\gamma|k\rangle|0\rangle + \sin\gamma|k\rangle|1\rangle)$

5

$y_j = 1$ when $j_{n+1} = 0$ and $x_{j_1 \ldots j_n} = 1$

Guaranteed that we will have only one such index.

so, $\cancel{\sin^2\gamma}$ $\frac{\cos^2\gamma}{2^n}$ = Probability for $y_j = 1$

7) Initially assume some index as the target index (i.e, minimum value attaining at that index and store that value.

Use Grover search algo to find any other index so that value at the new index is less ~~that~~ than the stored value.

3

If such a value is found you can update the ~~new~~ stored value, and repeat the grover's algorithm.

If no such value is found, re-run it in case algo failed $\leq 1/3$ probability.

Since in worst case we may update the stored value upto $O(\log N)$ times, the total queries is $O(\sqrt{N}\log N)$.