CPSC 436Q: Homework 3

Due on Gradescope by 11:59pm on November 25, 2024

Rules.

1. Please try to solve the problems yourself first. If you get stuck, you may *consult* any resources (books, internet, peers, office hours, etc.) for solutions. Provided you *acknowledge* these resources, no marks will be deducted. However, you must write up your own solution *independently*, using your own words.[1]

2. Please write legibly, work that is illegible will be marked as incorrect. Latex is strongly recommended for legibility. (I also recommend using `https://www.overleaf.com/` if you're new to Latex.)

3. All answers should be justified.

4. If you spot any mistakes, please email me at `wdaochen@cs.ubc.ca`. Any corrections will be announced on Piazza.

5. The total number of points for non-bonus questions is $T = 32$. Credit policy for the bonus questions: suppose you receive x points for the bonus questions and y points for the non-bonus questions, then the total number of points you receive for this homework is $\min(x + y, T)$.

Homework

1. **Consolidation of lecture material.**

   (a) **(2 points)** Linear algebra in $\mathbb{F}_2^k$. Show that the following three vectors *are* linearly independent as vectors in $\mathbb{R}^3$ (so span three dimensions) but *are not* linear independent as vectors in $\mathbb{F}_2^3$ (so span less than three dimensions). Write down the span of these vectors as vectors in $\mathbb{F}_2^3$.

$$\begin{pmatrix}1\\1\\0\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}, \begin{pmatrix}1\\0\\1\end{pmatrix}. \tag{1}$$

(Recall that in the analysis of Simon's algorithm, linear algebra is done in $\mathbb{F}_2^k$. While most aspects of linear algebra in $\mathbb{F}_2^k$ and $\mathbb{R}_2^k$ are the same, e.g., row-rank = column rank and the rank-nullity theorem, this exercise shows that there can be subtle differences.)

We show linear independence by contradiction. Suppose first there exists $a, b, c \in \mathbb{R}$ not all 0 such that

$$a\begin{pmatrix}1\\1\\0\end{pmatrix} + b\begin{pmatrix}0\\1\\1\end{pmatrix} + c\begin{pmatrix}1\\0\\1\end{pmatrix} = \mathbf{0}.$$

Then $a = -c$ (from the first entry), $a + b = -c + b = 0$ (from the second entry), and $b + c = 0 = -c + b$ (from the last entry). So we have $c = -c \implies c = 0$. Thus we also have $a = -c = 0$ and $-c + b = 0 + b = b = 0$, which is a contradiction since a, b, c are not all 0. So the vectors are linearly independent in $\mathbb{R}^3$. In contrast, in $\mathbb{F}_2^3$ we have

$$\begin{pmatrix}1\\1\\0\end{pmatrix} + \begin{pmatrix}0\\1\\1\end{pmatrix} + \begin{pmatrix}1\\0\\1\end{pmatrix} = \begin{pmatrix}1\\0\\1\end{pmatrix} + \begin{pmatrix}1\\0\\1\end{pmatrix} = \mathbf{0}$$

so we have found a nontrival linear combination of the vectors that sum to $\mathbf{0}$. Thus, the vectors are not linearly independent in $\mathbb{F}_2^3$. The span of the vectors in $\mathbb{F}_2^3$ is

$$\left\{ \begin{pmatrix}1\\1\\0\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}, \begin{pmatrix}1\\0\\1\end{pmatrix}, \begin{pmatrix}0\\0\\0\end{pmatrix} \right\}$$

which can be obtained by permutating the linear combinations.

[1] GenAI tools like ChatGPT can occasionally solve these problems correctly. Like other resources, if you use it, please verify and understand its solution first. Also remember, you will not have access to any resources other than a pen in the final exam.

(b) **(2 points)** Show that the quantum Fourier transform QFT_M is unitary for any $M \in \mathbb{N}$. That is, show

$$\mathrm{QFT}_M^\dagger\,\mathrm{QFT}_M = \mathbb{1}_M = \mathrm{QFT}_M\,\mathrm{QFT}_M^\dagger\,. \tag{2}$$

First note that $\mathrm{QFT}_1 = \mathbb{I}_1$ so the statement is clearly true. So we assume $M > 1 \in \mathbb{N}$. By multiplying QFT by the computational basis vectors, we can use the definition of QFT to construct its matrix representation. By symmetry, we can represent QFT_M and its transpose each in row form and column form.

$$QFT_M = \frac{1}{\sqrt{M}}\begin{bmatrix}\sum_{k=0}^{M-1}|k\rangle & \sum_{k=0}^{M-1}e^{\frac{2\pi ik\times 1}{M}}|k\rangle & \sum_{k=0}^{M-1}e^{\frac{2\pi ik\times 2}{M}}|k\rangle & \dots & \sum_{k=0}^{M-1}e^{\frac{2\pi ik\times(M-1)}{M}}|k\rangle\end{bmatrix}$$

$$= \frac{1}{\sqrt{M}}\begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 1 & e^{\frac{2\pi i\times 1\times 1}{M}} & e^{\frac{2\pi i\times 1\times 2}{M}} & \dots & e^{\frac{2\pi i\times 1\times(M-1)}{M}}\\ 1 & e^{\frac{2\pi i\times 2\times 1}{M}} & e^{\frac{2\pi i\times 2\times 2}{M}} & \dots & e^{\frac{2\pi i\times 2\times(M-1)}{M}}\\ \dots & \dots & \dots & \dots & \dots\\ 1 & e^{\frac{2\pi i\times(M-1)\times 1}{M}} & e^{\frac{2\pi i\times(M-1)\times 2}{M}} & \dots & e^{\frac{2\pi i\times(M-1)\times(M-1)}{M}}\end{bmatrix}$$

$$= \frac{1}{\sqrt{M}}\begin{bmatrix}\sum_{k=0}^{M-1}\langle k|\\ \sum_{k=0}^{M-1}e^{\frac{2\pi ik\times 1}{M}}\langle k|\\ \sum_{k=0}^{M-1}e^{\frac{2\pi ik\times 2}{M}}\langle k|\\ \dots\\ \sum_{k=0}^{M-1}e^{\frac{2\pi ik\times(M-1)}{M}}\langle k|\end{bmatrix}$$

Similarly, the conjugate transpose is

$$QFT_M^\dagger = \frac{1}{\sqrt{M}}\begin{bmatrix}\sum_{k=0}^{M-1}\langle k|\\ \sum_{k=0}^{M-1}e^{-\frac{2\pi ik\times 1}{M}}\langle k|\\ \sum_{k=0}^{M-1}e^{-\frac{2\pi ik\times 2}{M}}\langle k|\\ \dots\\ \sum_{k=0}^{M-1}e^{-\frac{2\pi ik\times(M-1)}{M}}\langle k|\end{bmatrix}$$

$$= \frac{1}{\sqrt{M}}\begin{bmatrix}\sum_{k=0}^{M-1}|k\rangle & \sum_{k=0}^{M-1}e^{-\frac{2\pi ik\times 1}{M}}|k\rangle & \sum_{k=0}^{M-1}e^{-\frac{2\pi ik\times 2}{M}}|k\rangle & \dots & \sum_{k=0}^{M-1}e^{-\frac{2\pi ik\times(M-1)}{M}}|k\rangle\end{bmatrix}$$

So

$$QFT_M QFT_M^\dagger$$

$$= \frac{1}{M}\begin{bmatrix}\sum_{k=0}^{M-1}\langle k|\\ \sum_{k=0}^{M-1}e^{\frac{2\pi ik\times 1}{M}}\langle k|\\ \dots\\ \sum_{k=0}^{M-1}e^{\frac{2\pi ik\times(M-1)}{M}}\langle k|\end{bmatrix}\begin{bmatrix}\sum_{k=0}^{M-1}|k\rangle & \sum_{k=0}^{M-1}e^{-\frac{2\pi ik\times 1}{M}}|k\rangle & \dots & \sum_{k=0}^{M-1}e^{-\frac{2\pi ik\times(M-1)}{M}}|k\rangle\end{bmatrix}$$

The diagonal entries become M due to orthogonality of basis and the exponential constants canceling out. Consider a general non-diagonal entry. For $l_1, l_2 \in \{0, ..., M-1\}, l_1 \neq l_2$, a general non-diagonal entry can be represented as the product :

$$\Big(\sum_{k=0}^{M-1}e^{\frac{2\pi ik\times l_1}{M}}\langle k|\Big)\Big(\sum_{k=0}^{M-1}e^{-\frac{2\pi ik\times l_2}{M}}|k\rangle\Big)$$

$$= \sum_{k=0}^{M-1}e^{\frac{2\pi ik\times(l_1-l_2)}{M}}$$

We recognize this as a finite geometric sequence with $r = e^{\frac{2\pi i\times(l_1-l_2)}{M}} \neq 1$ (since $l_1 \neq l_2$ and $M > 1$), giving us

$$= \frac{1-(e^{\frac{2\pi i\times(l_1-l_2)}{M}})^M}{1-r} = \frac{1-e^{2\pi i\times(l_1-l_2)}}{1-r} = \frac{1-1}{1-r} = 0$$

since $l_1 - l_2 \in \mathbb{Z}$. So $QFT_M QFT_M^\dagger = \frac{1}{M}\mathrm{diag}(M, M, ..., M) = I_M$ as required.

Next,

$$QFT_M^\dagger QFT_M$$

$$= \frac{1}{M} \begin{bmatrix} \sum_{k=0}^{M-1} \langle k| \\ \sum_{k=0}^{M-1} e^{-\frac{2\pi i k \times 1}{M}} \langle k| \\ ... \\ \sum_{k=0}^{M-1} e^{-\frac{2\pi i k \times (M-1)}{M}} \langle k| \end{bmatrix} \begin{bmatrix} \sum_{k=0}^{M-1} |k\rangle & \sum_{k=0}^{M-1} e^{\frac{2\pi i k \times 1}{M}} |k\rangle & ... & \sum_{k=0}^{M-1} e^{\frac{2\pi i k \times (M-1)}{M}} |k\rangle \end{bmatrix}$$

Again, the diagonal entries become M due to the orthogonality of basis and the exponential constants canceling out. The non-diagonal entries are also zero following a similar argument as with the $QFT_M QFT_M^\dagger$ case above. So $QFT_M^\dagger QFT_M = \frac{1}{M}\text{diag}(M, M, ..., M) = I_M$ as required.

2. **Improving the period finding algorithm.**

In fact, the quantum query complexity of finding the period r of a string x with symbols in $\mathbb{Z}_N$ is $O(1)$, i.e., constant, which is better than the $O(\ln\ln(N))$ we derived in class. We will walk through how this works in this problem. (Based on Problem 8 from Assignment 6 of Ryan O'Donnell's 2018 course.)

Recall that the main subroutine of the quantum algorithm for period finding generates a number z of the form $j \cdot 2^n/r$ for some j uniformly at random from $\{0, 1, \ldots, r-1\}$. *Throughout this question, assume that r divides 2^n as in class.*

By rejecting $z = 0$, we can readily modify the subroutine to generate a number z of the form $j \cdot 2^n/r$ for some j uniformly at random from $\{1, \ldots, r-1\}$. This would only incur a small constant overhead since $z = 0$ occurs with probability $1/r$ (which is $\leq 1/100$ if we wlog assume $r \geq 100$ as in class).

In the following, all probabilities are over j_1, j_2 each independently chosen uniformly at random from $\{1, \ldots, r-1\}$.

(a) **(2 points)** Show that for any fixed prime $p \in \mathbb{N}$,

$$\Pr[p \text{ divides both } j_1 \text{ and } j_2] \leq \frac{1}{p^2}. \tag{3}$$

We note that for prime p to divide some number α, it is necessary that $\alpha \equiv 0 \mod p$ (α must be in this specific equivalent class). We also note for any r, since the range $\{1, ..., r-1\}$ starts with the equivalent class $1 \mod p$, at most $\frac{1}{p}$ of the integers in the range can be in the equivalent class $0 \mod p$. So at most $\frac{1}{p}$ of the integers in $\{1, ..., r-1\}$ can satisfy the necessary condition for p to divide α. Since α is chosen randomly and uniformly from $\{1, ..., r-1\}$, we have that the probability that p divides α is at most $\frac{1}{p}$. Since j_1, j_2 are chosen independently, we have

$$\Pr[p \text{ divides } j_1, j_2] = \Pr[p \text{ divides } j_1] \times \Pr[p \text{ divides } j_2] \leq \frac{1}{p} \times \frac{1}{p} = \frac{1}{p^2}$$

where we take α to be j_1 and j_2, as required.

(b) **(2 points)** Show that

$$\Pr[j_1 \text{ and } j_2 \text{ are not coprime}] \leq \sum_{p \text{ prime}} \frac{1}{p^2}, \tag{4}$$

where the second infinite sum is over all primes $p \in \mathbb{N}$. We first note that j_1 and j_2 are not coprime iff some prime number divides both j_1 and j_2. To see this, note we have by definition, j_1 and j_2 are not coprime iff their GCD is not 1. This is true in turn iff there exists some common factor ($\neq 1$) that divides j_1 and j_2, which is true iff there exists some common prime factor between j_1 and j_2 (prime factorize GCD(j_1, j_2) to obtain such a factor).

Now, we know from a) that the probability that a prime p divides j_1 and j_2 chosen randomly, uniformly, and independently from $\{1, ..., r-1\}$ is upper bounded by $\frac{1}{p}$. So we have

$$\Pr[j_1 \text{ and } j_2 \text{ are not coprime}] = \Pr[\exists \text{ prime } p \text{ st } p|j_1 \text{ and } p|j_2] \leq \sum_{p \text{ prime}} \frac{1}{p^2}$$

with the last equality holding because of union bound, as required.

(c) **(2 points)** Show that

$$\sum_{p \text{ prime}} \frac{1}{p^2} \leq 0.99. \tag{5}$$

[Hint: you may use any of the results/methods in Basel problem Wiki concerning a different but similar sum.] Note since the set of primes are a strict subset of $\mathbb{N}\backslash\{1\}$:

$$\sum_{p \text{ prime}} \frac{1}{p^2} < \sum_{n \in \mathbb{N}\backslash\{1\}} \frac{1}{n^2} = \sum_{n \in \mathbb{N}} \frac{1}{n^2} - 1 = \frac{\pi^2}{6} - 1 \approx 0.644 \leq 0.99$$

with

$$\sum_{n \in \mathbb{N}} \frac{1}{n^2} = \frac{\pi^2}{6}$$

being a well-known fact from the Basel problem Wiki.

(d) **(4 points)** Using the previous parts, explain how to *modify* the last part of the period-finding algorithm from class (where we repeated the subroutine $10000 \ln \ln(N)$ times) so that the number of repeats (and hence queries) becomes *constant* yet the algorithm is still *correct* with probability at least 2/3. (You may assume the subroutine has already been modified not to generate $z = 0$ because this only incurs a small constant overhead, as explained above.) [Hint: the notion of gcd may be useful. For $x, y \in \mathbb{N}$, $\gcd(x, y)$ is the greatest common divisor of x and y, e.g., $\gcd(30, 24) = 6$, $\gcd(8, 9) = 1$, $\gcd(7, 21) = 7$.]

As per instruction, we recall that the main subroutine of the alg generates a number z of the form $j \times 2^n/r$ for some j uniformly random from $\{1, ..., r-1\}$ (we exclude 0). Then, we consider the procedure (initialize $r^* = 0$):

i. each iteration, prepare and repeating the main subroutine twice, yielding two outcomes z_1 and z_2. Each with form $z_1 = \frac{j_1 2^n}{r}, z_2 = \frac{j_2 2^n}{r}$, $j_1, j_2 \in \{0, ..., r-1\}$

ii. Find the $gcd(z_1, z_2)$. Let $r' = \frac{2^n}{gcd(z_1,z_2)}$. Update $r^* = \max(r^*, r')$.

iii. repeat 1)-2) for 10000 iterations.

iv. output $r = r^*$

By 2a)-2c), we have

$$\Pr[j_1 \text{ and } j_2 \text{ are coprime}] = 1 - \Pr[j_1 \text{ and } j_2 \text{ are not coprime}] \geq 1 - \sum_{p \text{ prime}} \frac{1}{p^2} \geq 0.01$$

If j_1 and j_2 are coprime, it is clear that the gcd of z_1 and z_2 would be $2^n/r$ (can prime factorize to see this). Thus

$$r' = \frac{2^n}{gcd(z_1, z_2)} = r. \tag{1}$$

If not, then their gcd would be of form $2^n p/r$ for some $p > 1 \in \mathbb{N}$. Then $r' = \frac{2^n}{p \times gcd(z_1,z_2)} < r$ (hence we will compute a value smaller than the actual r, so we update r^* if $r' > r^*$). So

$$\Pr[\text{success (1 iter)}] = \Pr[z_1 = y_{j_1}, z_2 = y_{j_2}] \Pr[j_1 \text{ and } j_2 \text{ are coprime}] \geq 0.495^2 \times 0.01 = 0.00245025$$

$$\Pr[\text{failure (1 iter)}] = 1 - \Pr[\text{success (1 iter)}] = 1 - 0.00245025 = 0.99754975 \leq 0.998$$

Finally, we note that if we succeed for one iteration, then we are done by eq(1). So over 10000 iterations, the probability of failure is the probability of failing every round, which is $0.998^{10000} = 2.02028609 \times 10^{-9} \leq 1/3$. So indeed we succeed in 10000 iterations of the algorithm (20000 iterations of the main subroutine since each "iteration" is a pair of subroutines, which is still constant though) with a probability $\geq 2/3$ and we are done.

3. **Running Shor's algorithm yourself.** Recall the description of Shor's algorithm from the bottom of pg. 31 of my notes involving five steps. For *small* N, we can run it ourselves. In this exercise, take N to be 21.

(a) **(3 points)** Run the algorithm supposing that you chose $a = 2$ at the third step. Describe what happens at each step. You do *not* necessarily have to perform your computations using the red text in my notes. In the fourth step, you should compute $\text{ord}_N(a)$ using any (classical) method you can think of, e.g., brute force is fine.[2]

i. $N = 21$ is not even

ii. N is not a kth power. $\lceil \log_2(N) \rceil = 5$ so we only have to check up til $k = 5$. 21 is not a kth root because:

$$k = 2 : 4^2 = 16 < 21, 5^2 = 25 > 21$$

$$k = 3 : 2^3 = 8 < 21, 3^3 = 27 > 21$$

$$k = 4 : 2^4 = 16 < 21, 3^4 = 81 > 21$$

$$k = 5 : 2^5 = 32 > 21$$

iii. $a = 2$ so I want to find $gcd(2, 21)$. Clearly the gcd is 1 since 21 is not even.

[2] Of course, this step wouldn't be efficient classically as N gets large!

iv.

$$2^1 \not\equiv 1 \mod 21$$
$$2^2 = 4 \not\equiv 1 \mod 21$$
$$...$$
$$2^5 = 32 \not\equiv 1 \mod 21$$
$$2^6 = 64 \equiv 1 \mod 21$$

So $r = 6$. So we are in the lucky case.

v. $gcd(a^{r/2} - 1 \mod N, N) = gcd(7, 21) = 7$. So we output 7 since $7 > 1$.

(b) **(3 points)** Repeat the above but now supposing that you chose $a = 5$ at the third step.

i. $N = 21$ is not even

ii. N is not a kth power, as shown in part a).

iii. $a = 5$ so I want to find $gcd(5, 21)$. Use Euclidean algorithm.

$$21 = 5 \times 4 + 1$$
$$5 = 1 \times 5 + 0$$

so the gcd is 1.

iv.

$$5^1 \not\equiv 1 \mod 21$$
$$5^2 = 4 \not\equiv 1 \mod 21$$
$$5^3 = 125 \equiv 20 \not\equiv 1 \mod 21$$
$$5^4 = 625 \equiv 16 \not\equiv 1 \mod 21$$
$$5^5 = 3125 \equiv 17 \not\equiv 1 \mod 21$$
$$5^6 = 15625 \equiv 1 \mod 21$$

So $r = 6$. So we are in the lucky case.

v. $gcd(a^{r/2} - 1 \mod N, N) = gcd(124, 21)$. Use Euclidean algorithm.

$$124 = 21 \times 5 + 19$$
$$21 = 1 \times 19 + 2$$
$$19 = 9 \times 2 + 1$$
$$2 = 2 \times 1 + 0$$

so the $gcd(124, 21) = 1$. So we output "don't know".

4. **(4 points)** Show that $\text{Adv}(f) \geq \sqrt{s(f)}$.

First, suppose $s(f) = 0$. Then we want to show $\text{Adv}(f) \geq 0$. This is trivial because some Γ_i will have norm equal to We let $s(f) = s_x(f) = k$ for some string x. It suffices to consider the function on the partial input S where S consists of x and x^i for $i \in [n]$ where x^i denotes x with the ith bit flipped (by Andrew Childs 22.3 and Piazza post #37. Then one possible adversary matrix is

$$\Gamma = \begin{pmatrix} 0 & y_1 & y_2 & ... & y_n \\ y_1 & 0 & 0 & ... & 0 \\ y_2 & 0 & 0 & ... & 0 \\ ... & ... & ... & ... & ... \\ y_n & 0 & 0 & ... & 0 \end{pmatrix}$$

where $y_i = [f(x^i) == f(x)]$. We note that by definition of $s(f)$, exactly k of our y_is are non-zero. Then Γ_is have form

$$\begin{pmatrix} 0 & ... & y_i & ... & 0 \\ ... & ... & 0 & ... & ... \\ y_i & 0 & 0 & 0 & 0 \\ ... & ... & 0 & ... & ... \\ 0 & ... & 0 & ... & 0 \end{pmatrix}$$

where y_i is 1 or 0. We recall that the adversary bound takes the maximum over all adversary matrices Γ. So

$$Adv(f) \geq \frac{||\Gamma||}{\max_{i\in[n]} ||\Gamma_i||}$$

Now, we note that $||\Gamma_i||$ is either 0 or 1. To see this, we note that if $y_i = 0$, $\Gamma_i = \mathbf{0} \implies ||\Gamma_i|| = 0$. Otherwise, if $y_i = 1$, then

$$||\Gamma_i|| = \max_{||v||=1} ||\Gamma_i v|| = ||\begin{pmatrix} v_i \\ 0 \\ \dots \\ v_1 \\ \dots \\ 0 \end{pmatrix}|| \leq ||v|| = 1.$$

We also have $||\Gamma_i e_i|| = 1$. Thus $||\Gamma_i|| = 1$. Therefore, $\max_{i\in[n]} ||\Gamma_i|| = 1$, giving us

$$Adv(f) \geq ||\Gamma||$$

Now, we follow the approach from Andrew Childs 22.3

$$\Gamma^2 = \begin{pmatrix} k & 0 & 0 & \dots & 0 \\ 0 & y_{11} & y_{12} & \dots & y_{1n} \\ 0 & y_{21} & \dots & \dots & y_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & y_{n1} & \dots & \dots & y_{nn} \end{pmatrix}$$

where $y_{ij} = 1$ if $y_i = y_j = 1$, otherwise 0. Note $||\Gamma^2 e_1|| = k$, so by definition $||\Gamma^2|| = ||\Gamma||^2 \geq k \implies ||\Gamma|| \geq \sqrt{k}$ (note $k \geq 1$, and we have shown the first equality in HW1). So indeed

$$Adv(f) \geq ||\Gamma|| \geq \sqrt{k} = \sqrt{s(f)}$$

as required.

5. **The five-qubit code.** *You can do this question before we cover quantum error correction.* (Based on Problem 4 from Assignment 5 of Andrew Childs's 2019 course.) Consider the following two five-qubit states:

$$\begin{aligned} |0_L\rangle := \frac{1}{4}(&|00000\rangle \\ &+ |10010\rangle + |01001\rangle + |10100\rangle + |01010\rangle + |00101\rangle \\ &- |11000\rangle - |01100\rangle - |00110\rangle - |00011\rangle - |10001\rangle \\ &- |01111\rangle - |10111\rangle - |11011\rangle - |11101\rangle - |11110\rangle\,) \end{aligned}$$

and

$$\begin{aligned} |1_L\rangle := \frac{1}{4}(&|11111\rangle \\ &+ |01101\rangle + |10110\rangle + |01011\rangle + |10101\rangle + |11010\rangle \\ &- |00111\rangle - |10011\rangle - |11001\rangle - |11100\rangle - |01110\rangle \\ &- |10000\rangle - |01000\rangle - |00100\rangle - |00010\rangle - |00001\rangle\,). \end{aligned}$$

(a) **(4 points)** Show that $|0_L\rangle$ and $|1_L\rangle$ are eigenstates (i.e., eigenvectors) with eigenvalue $+1$ of each of the following four matrices:

$$\begin{array}{ccccccccccc} X & \otimes & Z & \otimes & Z & \otimes & X & \otimes & I & , \\ I & \otimes & X & \otimes & Z & \otimes & Z & \otimes & X & , \\ X & \otimes & I & \otimes & X & \otimes & Z & \otimes & Z & , \\ Z & \otimes & X & \otimes & I & \otimes & X & \otimes & Z & , \end{array} \tag{6}$$

where

$$I := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \text{and} \quad Z := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{7}$$

[Hint: you can show this without explicitly checking every case.]

We start with the first matrix. For simplicity, we let M_1, M_2, M_3, M_4 denote the four matrices respectively. Using the mixed product property, we have (M_1 flips the first and fourth bit of each basis and multiplies each basis $|i_1i_2i_3i_4i_5\rangle$ by $(-1)^{i_2+i_3}$)

$$\begin{aligned}
M_1\left|0_L\right\rangle = \frac{1}{4}(&M_1\left|00000\right\rangle \\
&+M_1\left|10010\right\rangle + M_1|01001\rangle + M_1\left|10100\right\rangle + M_1|01010\rangle + M_1\left|00101\right\rangle \\
&-M_1\left|11000\right\rangle - M_1|01100\rangle - M_1\left|00110\right\rangle - M_1|00011\rangle - M_1\left|10001\right\rangle \\
&-M_1\left|01111\right\rangle - M_1|10111\rangle - M_1\left|11011\right\rangle - M_1|11101\rangle - M_1\left|11110\right\rangle) \\
&= \frac{1}{4}(\left|10010\right\rangle \\
&\quad +\left|00000\right\rangle - \quad \left|11011\right\rangle - \left|00110\right\rangle - \quad \left|11000\right\rangle - \left|10111\right\rangle \\
&\quad +\left|01010\right\rangle - \quad \left|11110\right\rangle + \left|10100\right\rangle - \quad \left|10001\right\rangle - \left|00011\right\rangle \\
&\quad -\left|11101\right\rangle + \quad \left|00101\right\rangle + \left|01001\right\rangle - \quad \left|01111\right\rangle - \left|01100\right\rangle) \\
= \left|0_L\right\rangle&
\end{aligned}$$

We also have

$$\begin{aligned}
M_1\left|1_L\right\rangle = \frac{1}{4}(&M_1\left|11111\right\rangle \\
&+M_1\left|01101\right\rangle + M_1|10110\rangle + M_1\left|01011\right\rangle + M_1|10101\rangle + M_1\left|11010\right\rangle \\
&-M_1\left|00111\right\rangle - M_1|10011\rangle - M_1\left|11001\right\rangle - M_1|11100\rangle - M_1\left|01110\right\rangle \\
&-M_1\left|10000\right\rangle - M_1|01000\rangle - M_1\left|00100\right\rangle - M_1|00010\rangle - M_1\left|00001\right\rangle\,) \\
&= \frac{1}{4}(\left|01101\right\rangle \\
&\quad +\left|11111\right\rangle - \quad \left|00100\right\rangle - \left|11001\right\rangle - \quad \left|00111\right\rangle - \left|01000\right\rangle \\
&\quad +\left|10101\right\rangle - \quad \left|00001\right\rangle + \left|01011\right\rangle - \quad \left|01110\right\rangle - \left|11100\right\rangle \\
&\quad -\left|00010\right\rangle + \quad \left|11010\right\rangle + \left|10110\right\rangle - \quad \left|10000\right\rangle - \left|10011\right\rangle\,) \\
= \left|1_L\right\rangle&
\end{aligned}$$

So we have that $|0_L\rangle$ and $|1_L\rangle$ are eigenstates of M_1 with eigenvalue $+1$ as required.
Now, we notice the cyclic pattern in our 4 matrices. Namely, we can interpret the effects of each one of five matrix in each of our 4 matrix as an action on a bit (because of mixed product property), and that if we shift these actions one spot to the right in our first matrix, we obtain the second. One spot to the right in our second matrix we obtain the third. Third we obtain Fourth. Let $>>$ denote this "shifting" procedure on $M_1, ..., M_3$. In this notation,

$$M_1 >> 1 = M_2, M_2 >> 1 = M_3, M_3 >> 1 = M_4$$

We additionally define shifting for quantum states:

$$\sum_{i_1=0}^{1}\sum_{i_2=0}^{1}\sum_{i_3=0}^{1}\sum_{i_4=0}^{1}\sum_{i_5=0}^{1}\alpha_{i_1i_2i_3i_4i_5}\left|i_1i_2i_3i_4i_5\right\rangle >> 1 = \sum_{i_1=0}^{1}\sum_{i_2=0}^{1}\sum_{i_3=0}^{1}\sum_{i_4=0}^{1}\sum_{i_5=0}^{1}\alpha_{i_1i_2i_3i_4i_5}\left|i_5i_1i_2i_3i_4\right\rangle.$$

We also define $<<$ as the inverse operation of $>>$ (defined only for $M_2, ..., M_4$). For M_2, M_3, M_4, we know $(M_i\left|x\right\rangle) << 1 = (M_i << 1)(\left|x\right\rangle << 1)$ since applying the action on every bit, then shifting the state, is the same as shifting the action to apply on bits one position shifted, and applying the actions on the shifted the state (notice the actions are applied on the same bits). Lastly, we note (can be easily checked) that $|0_L\rangle >> 1 = |0_L\rangle$, $|1_L\rangle >> 1 = |1_L\rangle$, $|0_L\rangle << 1 = |0_L\rangle$, $|1_L\rangle << 1 = |1_L\rangle$.
So overall, we have

$$M_2\left|0_L\right\rangle = ((M_2\left|0_L\right\rangle) << 1) >> 1 = (M_1(\left|0_L\right\rangle << 1)) >> 1 = (M_1\left|0_L\right\rangle) >> 1 = \left|0_L\right\rangle >> 1 = \left|0_L\right\rangle$$

$$M_2\left|1_L\right\rangle = ((M_2\left|1_L\right\rangle) << 1) >> 1 = (M_1(\left|1_L\right\rangle << 1)) >> 1 = (M_1\left|1_L\right\rangle) >> 1 = \left|1_L\right\rangle >> 1 = \left|1_L\right\rangle$$

$$M_3\left|0_L\right\rangle = ((M_3\left|0_L\right\rangle) << 1) >> 1 = (M_2(\left|0_L\right\rangle << 1)) >> 1 = (M_2\left|0_L\right\rangle) >> 1 = \left|0_L\right\rangle >> 1 = \left|0_L\right\rangle$$

$$M_3\left|1_L\right\rangle = ((M_3\left|1_L\right\rangle) << 1) >> 1 = (M_2(\left|1_L\right\rangle << 1)) >> 1 = (M_2\left|1_L\right\rangle) >> 1 = \left|1_L\right\rangle >> 1 = \left|1_L\right\rangle$$

$$M_4\left|0_L\right\rangle = ((M_4\left|0_L\right\rangle) << 1) >> 1 = (M_3(\left|0_L\right\rangle << 1)) >> 1 = (M_3\left|0_L\right\rangle) >> 1 = \left|0_L\right\rangle >> 1 = \left|0_L\right\rangle$$

$$M_4 |1_L\rangle = ((M_4 |1_L\rangle) << 1) >> 1 = (M_3(|1_L\rangle << 1)) >> 1 = (M_3 |1_L\rangle) >> 1 = |1_L\rangle >> 1 = |1_L\rangle$$

so indeed $|0_L\rangle, |1_L\rangle$ are eigenstates of M_1, M_2, M_3, M_4 with eigenvalue $+1$ as required.

(b) **(2 points)** Show that any pair of the the four matrices in eq. (6) commute. (We say two square matrices A, B of the same size commute if $AB = BA$.)

When we multiply any pair of the given four matrices, we have

$$(A_1 \otimes A_2 \otimes A_3 \otimes A_4 \otimes A_5)(B_1 \otimes B_2 \otimes B_3 \otimes B_4 \otimes B_5)$$

$$= A_1B_1 \otimes A_2B_2 \otimes A_3B_3 \otimes A_4B_4 \otimes A_5B_5$$

where A_i, B_i are some matrices in $\{I, X, Z\}$. Note $II = II, XX = XX, ZZ = ZZ, IX = X = XI, IZ = Z = ZI$ but

$$XZ = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, ZX = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = -XZ.$$

So if the number of pairs of A_iB_i that are ZX or XZ is even, we have for k even,

$$A_1B_1 \otimes A_2B_2 \otimes A_3B_3 \otimes A_4B_4 \otimes A_5B_5$$

$$= (-1)^k(B_1A_1 \otimes B_2A_2 \otimes B_3A_3 \otimes B_4A_4 \otimes B_5A_5)$$

$$= (B_1 \otimes B_2 \otimes B_3 \otimes B_4 \otimes B_5)(A_1 \otimes A_2 \otimes A_3 \otimes A_4 \otimes A_5)$$

Indeed when multiplying any pair of the four matrices, the number of pairs of A_iB_i that are XZ or ZX is even, so any pair of the four matrices commute and we are done.

(c) **(2 points)** Show that $X_L \coloneqq X \otimes X \otimes X \otimes X \otimes X$ and $Z_L \coloneqq Z \otimes Z \otimes Z \otimes Z \otimes Z$ satisfy

$$X_L |0_L\rangle = |1_L\rangle, \quad X_L |1_L\rangle = |0_L\rangle, \quad Z_L |0_L\rangle = |0_L\rangle, \quad \text{and} \quad Z_L |1_L\rangle = -|1_L\rangle. \tag{8}$$

By the mixed-product property, we know $X_L |0_L\rangle$ is flipping all the bits in the binary representation of the basis summed in $|0_L\rangle$. We can see clearly that flipping each bit of each basis summed in $|0_L\rangle$ gives $|1_L\rangle$. So indeed $X_L |0_L\rangle = |1_L\rangle$. Since X_L is self inverting $(X_LX_L = (XX) \otimes (XX) \otimes (XX) \otimes (XX) \otimes (XX) = I)$, we have

$$X_L |0_L\rangle = |1_L\rangle \implies X_LX_L |0_L\rangle = X_L |1_L\rangle \implies |0_L\rangle = X_L |1_L\rangle$$

as required.

Now, also using mixed-product property, we know that $Z_L |0_L\rangle$ is multiplying each summed basis by a factor of -1 for each 1 in the binary representation of the basis. In $|0_L\rangle$, each summed basis has an even number of 1s in its binary representation. Therefore, no scaling results for any basis in $|0_L\rangle$. Indeed $Z_L |0_L\rangle = |0_L\rangle$. Similarly, since each basis in $|1_L\rangle$ has an odd number of bits in its binary representation, Z_L scales each summed basis in $|1_L\rangle$ by -1. So overall, $Z_L |1_L\rangle = -Z_L |1_L\rangle$, as required.

6. **Bonus questions.**

(a) **No quantum advantage for parity.** The $\text{PARITY}_n\colon \{0,1\}^n \to \{0,1\}$ function is defined by

$$\text{PARITY}_n(x) = x_1 \oplus x_2 \oplus \cdots \oplus x_n. \tag{9}$$

(4 points) Show that $\text{Adv}(\text{PARITY}_n) = \Omega(n)$.

(b) **(4 points)** In class, we showed that the probability $P_{K,k}$ of K vectors, each chosen uniformly at random from $\mathbb{F}_2^k$, spanning k dimensions is at least $1 - 2^{k-K}$. When $K = k$, this bound is trivial. Derive an exact formula for $P_{k,k}$ and show that it is, in fact, at least $1/4$ for any $k \in \mathbb{N}$.