

# CMPT 476/776: Introduction to Quantum Algorithms

## Assignment 4

Due **March 26th, 2026 at 11:59pm on crowdmark**

Complete individually and submit in PDF format.

### Question 1 [7 points]: Coset states and Generalized Simon

Recall that the dot product on the vector space  $\mathbb{Z}_2^n$  is defined as  $x \cdot y = x_1y_1 \oplus x_2y_2 \oplus \dots \oplus x_ny_n$  where  $x = (x_1, x_2, \dots, x_n), y = (y_1, y_2, \dots, y_n) \in \mathbb{Z}_2^n$ . For any subspace  $S$  of  $\mathbb{Z}_2^n$ , define the orthogonal complement of  $S$  with respect to the dot product as

$$S^\perp = \{z \in \mathbb{Z}_2^n \mid s \cdot z = 0 \quad \forall s \in S\}.$$

1. Let  $|x + S\rangle = \frac{1}{\sqrt{|S|}} \sum_{s \in S} |x + s\rangle$  and show that

$$H^{\otimes n} |x + S\rangle = \sqrt{\frac{|S|}{2^n}} \sum_{z \in S^\perp} (-1)^{x \cdot z} |z\rangle$$

Hint: show that for any  $z \in \mathbb{Z}_2^n$ , either  $z \in S^\perp$  (i.e.  $z \cdot s = 0$  for all  $s \in S$ ) or  $z \cdot s = 0$  for exactly **half** the elements  $s \in S$ .

2. Show that Simon's algorithm can be generalized to solve the *Boolean hidden subgroup problem with no changes to the quantum part*. That is, given  $f : \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^n$  such that  $f(x) = f(y)$  if and only if  $x = y \oplus s$  for some  $s$  in a **non-trivial** linear subspace  $S$  of  $\mathbb{Z}_2^n$ , generalize Simon's algorithm to find a **basis** for  $S$ . You should sketch an algorithm in pseudo-code.

#### Solution.

1. First, we prove the result given in the hint. Let  $z \in \mathbb{Z}_2^n$  and suppose that  $z \notin S^\perp$ . We claim that  $z \cdot s = 0$  for exactly half the elements of  $S$  (and  $z \cdot s = 1$  for the other half). There are numerous ways to prove this. One way is to use a linear algebra approach. Consider the linear map  $\varphi : S \rightarrow \mathbb{Z}_2$  defined by  $\varphi(s) = z \cdot s$ . Since  $z \notin S^\perp$ , there exists some  $s' \in S$  such that  $z \cdot s' = 1$ . Hence  $\varphi$  is surjective and has rank 1. Denoting the dimension of  $S$  by  $d$ , the rank-nullity theorem then implies that the nullity, i.e. the dimension of the kernel of  $\varphi$ , is  $d - 1$ . In other words, the subspace of  $S$  consisting of elements  $s$  such that  $z \cdot s = 0$  has  $2^{d-1}$  elements, which is half of  $|S| = 2^d$ . So we're done.

Now, we have

$$\begin{aligned}
H^{\otimes n}|x + S\rangle &= \frac{1}{\sqrt{2^n|S|}} \sum_{s \in S} \sum_{z \in \mathbb{Z}_2^n} (-1)^{(x+s) \cdot z} |z\rangle \\
&= \frac{1}{\sqrt{2^n|S|}} \sum_{s \in S} \left( \sum_{z \in S^\perp} (-1)^{(x+s) \cdot z} |z\rangle + \sum_{z \notin S^\perp} (-1)^{(x+s) \cdot z} |z\rangle \right) \\
&= \frac{1}{\sqrt{2^n|S|}} \left( \sum_{z \in S^\perp} \sum_{s \in S} (-1)^{x \cdot z} |z\rangle + \sum_{z \notin S^\perp} \sum_{s \in S} (-1)^{x \cdot z} (-1)^{s \cdot z} |z\rangle \right) \\
&= \frac{1}{\sqrt{2^n|S|}} \left( \sum_{z \in S^\perp} (-1)^{x \cdot z} \sum_{s \in S} |z\rangle + \sum_{z \notin S^\perp} (-1)^{x \cdot z} \sum_{s \in S} (-1)^{s \cdot z} |z\rangle \right),
\end{aligned}$$

where in the third line we switch the summations and use the fact that  $s \cdot z = 0$  for  $z \in S^\perp$ ,  $s \in S$ . The first double sum now simplifies to  $\sum_{z \in S^\perp} (-1)^{x \cdot z} |S| |z\rangle$ . For the second double sum, we use the result we just proved. Looking at  $\sum_{s \in S} (-1)^{s \cdot z} |z\rangle$ , since  $z \notin S^\perp$ , half of the amplitudes will be  $-1$  and the other half will be  $1$ . Thus the terms cancel out.

Therefore, we're left with the state  $\sqrt{\frac{|S|}{2^n}} \sum_{z \in S^\perp} (-1)^{x \cdot z} |z\rangle$ , as required.

## 2. Example algorithm:

- 1:  $A \leftarrow [ \ ]$
- 2: **for**  $i$  from 0 to  $\approx n$  **do**
- 3:     apply  $H^{\otimes n} \otimes I^{\otimes n}$
- 4:     apply quantum oracle  $U_f$
- 5:     measure the last  $n$  qubits in computational basis
- 6:     apply  $H^{\otimes n} \otimes I^{\otimes n}$
- 7:     measure in computational basis to obtain output  $z \in S^\perp$
- 8:     append row  $z$  to  $A$
- 9: **end for**
- 10: find a basis for the kernel of  $A$

Steps 3 to 7 result in the state  $|x + S\rangle$  for some  $x$ . Then part 1 of this question shows that step 8 gives us a uniform superposition of states labelled by bit strings in  $S^\perp$ . We then sample until **with high probability** we have a basis for  $S^\perp$ , which we then use in step 12 to find  $S$ . Alternatively, if we know that  $\dim(S) = m$ , then by rank-nullity  $\dim(S^\perp) = n - m$ , so we only need to sample until we have  $n - m$  linearly independent samples to get a basis of  $S^\perp$  with 100% probability.

## Question 2 [3 points]: Factoring, classically

In this question we will factor the number 21 classically. You do not have to show your calculations and you may find it useful to use a calculator or program to calculate the GCD. If it were me, I would probably write a program to do it.

1. Compute the period of  $f(x) = 5^x \pmod{21}$  — that is, find the smallest integer  $r$  such that  $5^r \equiv 1 \pmod{21}$ .
2. Compute  $GCD(5^{r/2} + 1, 21)$ ,  $GCD(5^{r/2} - 1, 21)$ . What's the problem?
3. Now repeat steps 1 and 2 with  $f(x) = 2^x \pmod{21}$  to factor 21 into its prime factors.

**Solution.**

1.  $r = 6$ .
2.  $GCD(126, 21) = 21$ ,  $GCD(124, 21) = 1$ . So we have not found any non-trivial factors of 21.
3. We also have  $r = 6$ , so that we have  $GCD(9, 21) = 3$ ,  $GCD(7, 21) = 7$ . So both 3 and 7 are factors of 21 (and in fact  $3 \cdot 7 = 21$ ).

**Question 3 [4 points]: QFT or QFT<sup>-1</sup>?**

In class we've been pretty cavalier about whether we use QFT or the  $QFT^{-1} = QFT^\dagger$  in period finding and phase estimation. In this question we'll investigate why.

1. Determine what transformation is applied by  $QFT_{2^n}^2$  — that is, compute  $QFT_{2^n}(QFT_{2^n}|x\rangle)$  where  $x \in \{0, 1\}^n$ .
2. Use the result of the previous question to determine the **order** of the QFT (i.e. the minimal  $k \in \mathbb{Z}$  such that  $QFT^k = I$ ).
3. Now suppose you accidentally applied QFT when you should have applied  $QFT^{-1}$  and measured the result to get a bit string  $y \in \{0, 1\}^n$ . How could you **classically** recover from  $y$  the “correct” bit string  $z \in \{0, 1\}^n$  which you would have measured if you had instead applied  $QFT^{-1}$ ?

**Solution.**

1. Letting  $\omega = e^{\frac{2\pi i}{2^n}}$ , we have

$$\begin{aligned} QFT(QFT|x\rangle) &= QFT \left( \frac{1}{\sqrt{2^n}} \sum_{y \in \mathbb{Z}_{2^n}} \omega^{xy} |y\rangle \right) \\ &= \frac{1}{2^n} \sum_{y \in \mathbb{Z}_{2^n}} \omega^{xy} \sum_{z \in \mathbb{Z}_{2^n}} \omega^{yz} |z\rangle \\ &= \frac{1}{2^n} \sum_{z \in \mathbb{Z}_{2^n}} \sum_{y \in \mathbb{Z}_{2^n}} \omega^{(x+z)y} |z\rangle. \end{aligned}$$

Now  $\sum_{y \in \mathbb{Z}_{2^n}} \omega^{(x+z)y} = 0$  if and only if  $x + z \not\equiv 0 \pmod{2^n}$ . So the only terms that don't cancel are the ones where  $z = 2^n - x$ , in which case  $\omega^{x+z} = 1$ . Thus we get

$$QFT(QFT|x\rangle) = \frac{1}{2^n} \sum_{y \in \mathbb{Z}_{2^n}} |2^n - x\rangle = |2^n - x\rangle.$$

- By the previous question,  $QFT^2|x\rangle = |2^n - x\rangle$ , hence  $QFT^4|x\rangle = QFT^2|2^n - x\rangle = |2^n - (2^n - x)\rangle = |x\rangle$ . Note also that  $QFT^2 \neq I \neq QFT^3$ , in particular as  $QFT^3 = QFT^2 \cdot QFT$  is just  $QFT$  up to a permutation of  $\mathbb{Z}_{2^n}$ .
- Suppose we have the state  $|\psi\rangle = \sum_{i \in \mathbb{Z}_{2^n}} a_i |i\rangle$ , where the  $|i\rangle$  form a basis for the underlying Hilbert space. Using part 2,  $QFT^3 = QFT^{-1}$ , and in particular  $QFT^{-1} = QFT^2 \cdot QFT$ . The key observation is that  $QFT^2$  is really just a relabeling of the computational basis states  $x \mapsto 2^n - x$ , so we can apply this relabeling to the result  $y$  to give the corresponding measurement  $z = 2^n - y$  that we would have obtained if we had instead applied  $QFT^{-1}$ .

## Question 4 [5 points]: Modular multiplication

Recall that Shor's algorithm requires an (expensive) oracle for performing the classical function

$$x \mapsto a^x \pmod{M},$$

where  $a^x \pmod{M}$  can be written as a sequence of multiplications  $a^{x_0}(a^2)^{x_1} \dots (a^{2^{n-1}})^{x_{n-1}} \pmod{M}$ .

- Let  $M = 2^n$ . Give **classical** pseudo-code for adding two length  $n$ -binary integers mod  $2^n$ .
- Using addition mod  $2^n$  as a sub-routine, give pseudo-code for binary multiplication mod  $2^n$ .
- Give an expression in big-O notation for the complexity of modular exponentiation mod  $2^n$  using the sub-routines you developed in parts 1 and 2.

### Solution.

- This is just binary addition, which can be performed via long addition. Pseudo-code below, adding two length  $n$  binary integers  $A$  and  $B$  into a length  $n$  binary integer  $R$ 
  - $C \leftarrow 0, R \leftarrow [00 \dots 0]$
  - for**  $i$  from 0 to  $n - 1$  **do**
  - $R[i] \leftarrow A[i] \oplus B[i] \oplus C$
  - $C \leftarrow A[i] \cdot B[i] \oplus A[i] \cdot C \oplus B[i] \cdot C$  (majority of  $A[i], B[i]$ , and  $C$ )
  - end for**
  - return  $R$
- Likewise, this is just binary long multiplication, returning a length  $n$  binary integer as the result is taken mod  $2^n$ :
  - $R \leftarrow [00 \dots 0]$
  - for**  $i$  from 0 to  $n - 1$  **do**
  - if**  $A[i] == 1$  **then**
  - $R \leftarrow R + (B \ll i) \pmod{2^n}$  (left-shift  $B$  by  $i$  to multiply by  $2^i$ )
  - end if**
  - end for**
  - return  $R$

3. Noting that addition is  $O(n)$ , and the loop in the multiplier performs  $O(n)$  additions, the complexity of multiplication  $\text{mod } 2^n$  is  $O(n^2)$ . Finally, modular exponentiation  $|x\rangle|y\rangle \mapsto |x\rangle|y \cdot a^x \text{ mod } 2^n\rangle$  is decomposed into a sequence of  $n$  modular multiplications for a total of  $O(n^3)$  time, as

$$\begin{aligned} a^x &= a^{x_0+2x_1+\dots+2^{n-1}x_{n-1}} \\ &= a^{x_0}(a^2)^{x_1} \dots (a^{2^{n-1}})^{x_{n-1}} \\ &\equiv (a \text{ mod } 2^n)^{x_0}(a^2 \text{ mod } 2^n)^{x_1} \dots (a^{2^{n-1}} \text{ mod } 2^n)^{x_{n-1}} \text{ mod } 2^n. \end{aligned}$$

Note that the above decomposition requires pre-computing  $a^{2^i} \text{ mod } 2^n$  for each  $i \in [0, n-1]$ , so we also have to check the complexity of these classical pre-computations. In this case,  $a^{2^i} \equiv (\dots(a^2 \text{ mod } 2^n)^2 \text{ mod } 2^n)^2 \text{ mod } 2^n$ , so each  $a^{2^i} \text{ mod } 2^n$  can be computed classically in at most  $O(n^3)$  time. This is the technique known as *repeated squaring*.

## Question 5 [7 points]: Qutrit quantum computing

Much of quantum computation can be generalized to higher-dimensional **qudits**. Most gates we've seen have higher-dimensional generalizations, like the **Pauli gates**  $X, Y, Z$  and the Hadamard or **Fourier** gate  $H$ . In this question we will explore this notion briefly.

Consider a **qutrit**, which is a 3-dimensional quantum state — i.e. a unit vector in  $\mathbb{C}^3$ . As discussed in class, we denote the computational basis of  $\mathbb{C}^3$  as  $\{|0\rangle, |1\rangle, |2\rangle\}$ , or  $|x\rangle$  where  $x \in \mathbb{Z}_3$ , the integers mod 3. Denote the primitive third root of unity as  $\omega_3 = e^{2\pi i/3}$ . The Pauli  $X$  and  $Z$  operators on a qutrit can now be defined as

$$X = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \omega_3 & 0 \\ 0 & 0 & \omega_3^2 \end{bmatrix}$$

Likewise, the qutrit Hadamard gate can be defined as

$$H = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \omega_3 & \omega_3^2 \\ 1 & \omega_3^2 & \omega_3 \end{bmatrix}$$

1. Show that  $X$  and  $Z$  have order 3 (i.e.  $X^3 = Z^3 = I$ )
2. Show that  $XZ = \omega_3^2 ZX$ . Use this to calculate  $k$  (as a function of  $i$  and  $j$ ) such that  $X^i Z^j = \omega^k Z^j X^i$  for  $i, j \in \{0, 1, 2\}$ .
3. Show that  $H^\dagger Z H = X$ .
4. Compute the eigenvalues of  $X$  and give corresponding (unit) eigenvectors. Hint: recall the relationship between  $H$  and the eigenvectors of  $X$  in the qubit case.
5. Now show that Deutsch's algorithm generalizes to *qutrits*. Explicitly, given a function  $f : \mathbb{Z}_3 \rightarrow \mathbb{Z}_3$  promised to either be **constant** or **balanced** where balanced in this case means

for every  $y \in \mathbb{Z}_3$ , there exists exactly one  $x \in \mathbb{Z}_3$  such that  $f(x) = y$ , show that Deutsch's algorithm with the qutrit version of the  $H$  gate works the same way.

Hint: note that the qutrit  $H$  gate is the 3-dimensional QFT. That is,

$$H|x\rangle = \frac{1}{\sqrt{3}} \sum_{z \in \mathbb{Z}_3} \omega_3^{xz} |z\rangle.$$

**Solution.**

$$1. Z^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \omega_3^2 & 0 \\ 0 & 0 & \omega_3^4 \end{bmatrix} \neq I, Z^3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \omega_3^3 & 0 \\ 0 & 0 & \omega_3^6 \end{bmatrix} = I.$$

Alternatively, note that  $Z|x\rangle = \omega_3^x|x\rangle$  and proceed.

For  $X$ , we can work with the matrix form, or we can note that  $X|x\rangle = |x+1 \pmod 3\rangle$ , so  $X^2|x\rangle = |x+2 \pmod 3\rangle \Rightarrow X^2 \neq I$ ,  $X^3|x\rangle = |x\rangle \Rightarrow X^3 = I$ .

2. Given a computational basis state  $|x\rangle$ , we have  $XZ|x\rangle = \omega_3^x X|x\rangle = \omega_3^x |x+1 \pmod 3\rangle$ , and  $ZX|x\rangle = Z|x+1 \pmod 3\rangle = \omega_3^{x+1} |x+1 \pmod 3\rangle$ , hence  $XZ = \omega_3^2 ZX$ .

Starting with  $X^i Z^j$ , we commute each  $X$  through the  $Z^j$ . Commuting  $X$  through one  $Z$  introduces a phase of  $\omega_3^2$ , so commuting  $i$   $X$ 's, each through  $j$   $Z$ 's, results in a phase of  $\omega_3^{2ij}$ , giving  $k \equiv 2ij \pmod 3$ .

3.

$$\begin{aligned} H^\dagger ZH &= \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \omega_3^2 & \omega_3 \\ 1 & \omega_3 & \omega_3^2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \omega_3 & 0 \\ 0 & 0 & \omega_3^2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \omega_3 & \omega_3^2 \\ 1 & \omega_3^2 & \omega_3 \end{bmatrix} \\ &= \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & \omega_3^2 & \omega_3 \\ 1 & \omega_3 & \omega_3^2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ \omega_3 & \omega_3^2 & 1 \\ \omega_3^2 & \omega_3 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = X, \end{aligned}$$

using the fact that  $1 + \omega_3 + \omega_3^2 = 0$ .

4. From the previous part, we have  $H^\dagger Z = XH^\dagger$ . Using the hint, we note that if  $|\psi\rangle$  is an eigenvector of  $Z$  with eigenvalue  $\lambda$ , then  $XH^\dagger|\psi\rangle = H^\dagger Z|\psi\rangle = \lambda H^\dagger|\psi\rangle$ . Thus  $H^\dagger|\psi\rangle$  is an eigenvector of  $X$  with eigenvalue  $\lambda$ . We can read off the eigenvalues of  $Z$  as  $1, \omega_3$  and  $\omega_3^2$ . Using the fact that  $Z|x\rangle = \omega_3^x|x\rangle$  for any computational basis state  $|x\rangle$ , it follows that the corresponding eigenvectors of  $Z$  are  $|0\rangle, |1\rangle$  and  $|2\rangle$ .

Therefore, the eigenvectors and corresponding eigenvalues of  $X$  are

$$\begin{aligned} H^\dagger|0\rangle &= \frac{1}{\sqrt{3}}(|0\rangle + |1\rangle + |2\rangle), & \text{with eigenvalue } 1; \\ H^\dagger|1\rangle &= \frac{1}{\sqrt{3}}(|0\rangle + \omega_3^2|1\rangle + \omega_3|2\rangle), & \text{with eigenvalue } \omega_3; \\ H^\dagger|2\rangle &= \frac{1}{\sqrt{3}}(|0\rangle + \omega_3|1\rangle + \omega_3^2|2\rangle), & \text{with eigenvalue } \omega_3^2. \end{aligned}$$

5. Starting with the state  $|0\rangle|1\rangle$ , we apply  $H \otimes H$  to get

$$\frac{1}{3} \left( \sum_{x \in \mathbb{Z}_3} |x\rangle \right) \left( \sum_{y \in \mathbb{Z}_3} \omega_3^y |y\rangle \right) = \frac{1}{3} \sum_{x, y \in \mathbb{Z}_3} \omega_3^y |x\rangle |y\rangle.$$

Now we apply  $U_f$ :

$$\begin{aligned} \frac{1}{3} \sum_{x, y \in \mathbb{Z}_3} \omega_3^y |x\rangle |y + f(x)\rangle &= \frac{1}{3} \sum_{x \in \mathbb{Z}_3} |x\rangle (|f(x)\rangle + \omega_3 |1 + f(x)\rangle + \omega_3^2 |2 + f(x)\rangle) \\ &= \frac{1}{3} \sum_{x \in \mathbb{Z}_3} \omega_3^{-f(x)} |x\rangle (|0\rangle + \omega_3 |1\rangle + \omega_3^2 |2\rangle) \\ &= \frac{1}{3} \left( \sum_{x \in \mathbb{Z}_3} \omega_3^{-f(x)} |x\rangle \right) \left( \sum_{y \in \mathbb{Z}_3} \omega_3^y |y\rangle \right), \end{aligned}$$

where addition is modulo 3. Note that  $\omega_3^{-1} = \omega_3^2$ . Now we discard the second qubit and inspect the state of the first qubit. If  $f$  is constant, then up to global phase, we have  $\frac{1}{\sqrt{3}} \sum_{x \in \mathbb{Z}_3} |x\rangle = \frac{1}{\sqrt{3}} (|0\rangle + |1\rangle + |2\rangle)$ . If instead  $f$  is balanced, then up to global phase, the state is either  $\frac{1}{\sqrt{3}} (|0\rangle + \omega_3 |1\rangle + \omega_3^2 |2\rangle)$  or  $\frac{1}{\sqrt{3}} (|0\rangle + \omega_3^2 |1\rangle + \omega_3 |2\rangle)$ . If we now apply  $H^\dagger$  to this qubit, then we will end up with  $|0\rangle$  if  $f$  is constant, and either  $|1\rangle$  or  $|2\rangle$  if  $f$  is balanced. Thus, by taking a computational basis measurement, we can determine the nature of  $f$ . Alternatively, applying  $H^\dagger$  to  $\frac{1}{\sqrt{3}} \sum_{x \in \mathbb{Z}_3} \omega_3^{-f(x)} |x\rangle$  we get

$$\frac{1}{3} \sum_{x, z \in \mathbb{Z}_3} \omega_3^{-f(x) - xz} |z\rangle.$$

Looking at the state  $z = 0$  we see its amplitude is  $\frac{1}{3} \sum_{x \in \mathbb{Z}_3} \omega_3^{-f(x)} = 1$  if  $f(x)$  is constant, and otherwise the three distinct third roots of unity cancels out as in the binary case.