2026 Integrated Round **Quantum Computing**

Instructions

Note: This round is designed to not require all five days to complete.

Each problem's point value is written next to the problem; each part's point value is given in brackets next to the part. To receive full credit, the presentation must be orderly, clear, and concise. If a problem says "list" or "compute", you need not justify your answer. If a problem says "propose", you must provide motivation towards your reasoning but no proof/explanation is necessary. If a problem says "determine", "find", or "show", then you must show your work or explain your reasoning to receive full credit, although such explanations do not have to be lengthy. If a problem says "justify" or "prove", then you must prove your answer rigorously. Even if not proved, earlier numbered items may be used in solutions to later numbered items, but not vice versa.

Problem solutions must be submitted by LaTeX in the portal and/or by uploading files. General internet research (searching up general definitions and concepts) is permitted but must be cited. Researching specific properties of any question is not permitted. AI usage is not permitted. If you are unsure on whether or not a specific action/search is permitted or not, **assume it is not permitted**.

Note: Uploaded files do not save if you reload the website. Upload all your files right before submitting the round.

Introduction

You are part of a research team examining a sealed, highly classified piece of hardware. This hidden machine processes binary data.

It takes a string of n bits as input and spits out a single bit as output. In mathematical terms, it is a hidden function

$$f : \{0,1\}^n \to \{0,1\}.$$

Because the machine is sealed, you cannot look at its internal wiring. However, the manufacturer has promised you that the machine was built to respond only in one of two ways:

1. **Constant:** The machine ignores the input entirely. It will output the exact same value (either all 0s or all 1s) for every single one of the 2^n possible inputs.
2. **Balanced:** The machine is perfectly fair; half of all possible inputs will result in an output of 0, and the other half will result in an output of 1.

Your team's objective is as follows: **Determine which type of machine is sitting on the desk in front of you.**

To do this, you must query the machine by feeding it inputs and recording the outputs. Using a classical computer, you can only test **one** input at a time. Every test costs your team resources.

But your team has just been granted access to a prototype quantum chip. Instead of evaluating inputs one-by-one, the quantum chip uses the physics of waves and interference. By mapping the machine's outputs to positive and negative mathematical phases, it evaluates the global structure of the machine simultaneously, which causes the "wrong" possibilities to perfectly cancel each other out.

This Round explores the discrete mathematics behind how this is possible. This round is divided into four independent parts.

You do not need to solve Part I to begin Part II, nor do you need Part II to begin Part III.

Part I: Classical Computing

This section details the limitations of classical computation. In this system, you are required to submit an input x, and wait for the machine to return $f(x)$. Define every query to cost exactly 1 unit of time.

Let $N = 2^n$ be the total number of possible inputs.

> **Example: The $n = 2$ Machine**
>
> The inputs are strings of length 2. There are $N = 2^2 = 4$ possible inputs: $00, 01, 10, 11$.
> - If the machine is **Constant**, it might output: $f(00) = 0,\ f(01) = 0,\ f(10) = 0,\ f(11) = 0$.
> - If the machine is **Balanced**, it might output: $f(00) = 0,\ f(01) = 1,\ f(10) = 0,\ f(11) = 1$.

PROBLEM 1: **20 Points**

Suppose your research team requires 100% certainty before making a conclusion about the machine. For parts a, b, c, answer in a format similar to the one seen in the example above using lists of $f(...) = 0/1$

(a) **[5 pts]** For $n = 3$, list a set queried inputs and outputs for a machine $f(x)$ such that the team can conclude the machine is balanced, and compute the units of time used.

(b) **[5 pts]** For $n = 4$, list a set queried inputs and outputs for a machine $f(x)$ such that the team can conclude the machine is constant, and compute the units of time used.

(c) **[5 pts]** For $n = 7$, list a set of queried inputs and outputs for a function $f(x)$ such that would NOT allow the team to make a conclusion about the machine.

(d) **[5 pts]** Let $n = 2$ (so $N = 4$). Suppose you make two classical queries: $f(00)$ and $f(01)$. Both queries return 0. Determine whether the machine is constant, balanced, or unknown.

PROBLEM 2: **15 Points**

(a) **[5 pts]** For a general n-bit machine, suppose you query it k times, and every single query returns 0. Determine the minimum number of units of time k you must spend to absolutely guarantee the machine is constant, expressed in terms of n.

(b) **[10 pts]** Imagine the machine is not pre-programmed, but is instead controlled by a malicious adversary who decides the outputs on every turn you query to force you to make as many queries as possible, while still abiding by the Constant/Balanced rules. Justify your bound from part (a) for a worst-case scenario and describe how the adversary would respond.

PROBLEM 3: **25 Points**

For a modern 64-bit input ($n = 64$), the total number of inputs N can be treated as infinite. Reaching the deterministic bound from Problem 1 is not reasonable. Instead, your team decides to select inputs uniformly at random. If you query the machine k times and get the same output every time, you will guess "Constant."

(a) **[5 pts]** Let $n = 10$. You pick 3 distinct inputs entirely at random. If the machine is actually balanced, determine the probability that all 3 of your queries will return 0. (Leave your answer as a fraction).

(b) **[10 pts]** Find the general, closed-form expression for the probability that k distinct random queries all return 0, assuming the machine is actually balanced (for an arbitrary n).

(c) **[10 pts]** As N approaches infinity ($N \to \infty$), the probability of being "fooled" by a balanced machine after k identical outputs approaches a simple limit. Find this limit in terms of k. Using this limit, what is the minimum number of random queries k needed to be at least 99.9% confident that the machine is Constant?

PROBLEM 4: **20 Points**

Real-world classical computers suffer from thermal noise. Suppose the wire connecting your computer to the machine is degraded. Every time a bit travels down the wire, there is a probability $p = 0.1$ that the bit flips (a 0 becomes a 1, or a 1 becomes a 0).

(a) **[5 pts]** You query the *exact same* input x three times in a row to check for errors. The wire gives you the sequence: $[0, 0, 1]$. Assuming the true value of $f(x)$ is equally likely to be 0 or 1 before you started, compute the probability that the true output is actually 0 given the earlier sequence.

(b) **[15 pts]** Let $n = 3$. Assume the machine is equally likely to be Constant or Balanced. You query all distinct inputs exactly once. The wire gives you seven 0s and one 1.
Show that the machine is more likely to be constant than balanced by comparing specific values.

Part II: Quantum Computing

Note: This section does not require any results from Part I.

In this section, we treat every possible input x of our hidden machine as a formal "state", denoted by $|x\rangle$. If $x = 00$, then $|00\rangle$ is an envelope containing the string "00".

Because we cannot add regular numbers, we put them in these "envelopes".

Imagine we have a light switch. Usually, it can only be either on or off. A **Superposition** (see Glossary) is a "blur" of both of the states at the same time.

Mathematically, we denote this as a "weighted sum". If we have two states, $|0\rangle$ and $|1\rangle$, a superposition is just:

$$|\psi\rangle = |0\rangle + |1\rangle$$

Now, we introduce U_f, a machine operator based on a function f. This doesn't change any inputs but it denotes the phase of a given state.

Instead of returning a binary bit (0 or 1), this machine now embeds the answer into a **phase** (a positive or negative sign). Specifically, for an input x, the machine outputs:

$$(-1)^{f(x)}$$

Notice that if $f(x) = 0$, the output is $(-1)^0 = +1$. If $f(x) = 1$, the output is $(-1)^1 = -1$.

We write this mathematically as:

$$U_f|x\rangle = (-1)^{f(x)}|x\rangle$$

> **Example: Phase Mapping**
>
> For a 1-bit machine ($x \in \{0, 1\}$), the operation U_f maps the superposition $|\psi\rangle = |0\rangle + |1\rangle$ to:
>
> $$U_f(|\psi\rangle) = (-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle$$
>
> If f is constant and equal to 0, this results in $|0\rangle + |1\rangle$. If f is balanced, this results in $|0\rangle - |1\rangle$.

What does this example show? Notice that the "hidden information" about the machine (either constant or balanced) has been encoded into the signs of the superposition.

In classical computing, we would need two queries to know the answer. In this example, one operation gives the answer.

Note: In standard quantum mechanics, quantum states are "normalized" by a factor of $\frac{1}{\sqrt{2^n}}$ to ensure total probability equals 1. In this round, we omit this factor to simplify calculations, as we are primarily interested in the phase of the states rather than their absolute probability.

PROBLEM 5: **25 Points**

Consider a 2-bit system ($n = 2$). The inputs are $\{00, 01, 10, 11\}$.

(a) **[5 pts]** Compute the expression for the uniform superposition state $|\psi\rangle$.

(b) **[5 pts]** Suppose f is a balanced function defined by $f(00) = 0$, $f(01) = 1$, $f(10) = 0$, $f(11) = 1$. Compute $U_f|\psi\rangle$.

(c) **[5 pts]** Suppose f is a constant function defined by $f(00) = 0$, $f(01) = 0$, $f(10) = 0$, $f(11) = 0$. Compute $U_f|\psi\rangle$.

(d) **[5 pts]** Define the **Total Phase** of a superposition as the sum of its coefficients (e.g., for $a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$, the Total Phase is $a + b + c + d$). Calculate the Total Phase for your results in (b) and (c).

(e) **[5 pts]** Based on your answer in (d), propose a rule a team could use to identify the machine type.

PROBLEM 6: **10 Points**

Suppose we have a 3-bit system ($n = 3$). The inputs are $\{000, 001, \ldots, 111\}$. The uniform superposition is $|\psi\rangle = \sum_{x\in\{0,1\}^3} |x\rangle$.

(a) **[5 pts]** If the machine is balanced, calculate how many states $|x\rangle$ will have a coefficient of $+1$ and how many will have a coefficient of -1 after applying U_f.

(b) **[5 pts]** Calculate the Total Phase of $U_f|\psi\rangle$ for this balanced machine.

PROBLEM 7: **10 Points**

[10 pts] Prove that the operator U_f is linear. That is, show that for any two superpositions $|\psi_1\rangle$ and $|\psi_2\rangle$ and constants a, b, the relation $U_f(a|\psi_1\rangle + b|\psi_2\rangle) = aU_f|\psi_1\rangle + bU_f|\psi_2\rangle$ holds.

PROBLEM 8: **15 Points**

Let $f : \{0, 1\}^n \to \{0, 1\}$ be an arbitrary function. We define the Total Phase of the resulting state $U_f|\psi\rangle$ (where $|\psi\rangle$ is the uniform superposition of all 2^n inputs) as S.

(a) **[5 pts]** For a general n, find the maximum possible absolute value of the Total Phase $|S|$. Determine the condition on f of which this maximum occurs.

(b) **[5 pts]** If f is balanced, prove that the Total Phase S must be 0 for any n.

(c) **[5 pts]** For $n = 2$, find all possible values the Total Phase S can take across all possible functions f.

PROBLEM 9: **15 Points**

Often in quantum computing, we do not want to measure the "Total Phase" of a giant superposition because it is hard to calculate for large n. Instead, we use a technique called **Phase Kickback**, where we encode the result of $f(x)$ into the phase of a single bit.

(a) **[5 pts]** Consider the input state $|\phi\rangle = |0\rangle - |1\rangle$. Apply the operator U_f to this state (treating it as an operation on the single bit x). Expand and simplify $U_f(|0\rangle - |1\rangle)$, and compute the result.

(b) **[5 pts]** Prove that if $f(x) = 0$, the state remains $|\phi\rangle$. Prove that if $f(x) = 1$, the state becomes $-|\phi\rangle$.

(c) **[5 pts]** Justify why this is more powerful than the Total Phase sum: why is it better to have a single state change its global sign ($\pm$) rather than having to sum up 2^n different coefficients?

Part III: Deutsch's Algorithm

Note: This section uses the probability concepts from Part I and the phase-mapping math from Part II.

In Part II, we learned that the hidden machine can encode its outputs into mathematical signs. For a one-bit function $f : \{0, 1\} \to \{0, 1\}$, we start with the uniform superposition:

$$|\psi\rangle = |0\rangle + |1\rangle.$$

After applying the phase operator U_f, we get the resulting state:

$$U_f|\psi\rangle = (-1)^{f(0)}|0\rangle + (-1)^{f(1)}|1\rangle.$$

This state contains the information we want, but the information is hidden in the signs. If the signs are the same, then f is constant. If the signs are different, then f is balanced.

Before we extract this information, let's establish exactly why a classical computer struggles with this simple 1-bit machine.

15 Points

PROBLEM 10:

A classical computer can query only one input at a time.

(a) **[5 pts]** List one example of a constant function and one example of a balanced function that have the *exact same* value for $f(0)$.

(b) **[5 pts]** List one example of a constant function and one example of a balanced function that have the *exact same* value for $f(1)$.

(c) **[5 pts]** Show that no classical method using only a single query can ever determine whether f is constant or balanced.

The problem is that the quantum signs are not directly useful to us yet. If we measure the state right now, the $+/-$ phases disappear, and we just read a 0 or 1 with equal probability.

To extract the hidden relationship between the signs, we use an **interference filter** called the **Hadamard transform** (H). This filter forces matching signs to combine in one place and opposite signs to cancel out in another.

$$H|0\rangle = |0\rangle + |1\rangle, \qquad H|1\rangle = |0\rangle - |1\rangle.$$

20 Points

PROBLEM 11:

Hint: Recall from Part II that quantum operators are linear, meaning you can distribute them across a sum: $H(|0\rangle + |1\rangle) = H|0\rangle + H|1\rangle$.

(a) **[5 pts]** Compute the 2×2 matrix representation of H.

(b) **[5 pts]** Compute the expanded state of $H(|0\rangle + |1\rangle)$.

(c) **[5 pts]** Compute the expanded state of $H(|0\rangle - |1\rangle)$.

(d) **[5 pts]** Show how your answers to parts (b) and (c) demonstrate constructive and destructive interference.

---

15 Points

PROBLEM 12:

There are four possible one-bit hidden functions. Let Machine A be the Constant 0 function ($f(0) = 0, f(1) = 0$). Let Machine C be a balanced function ($f(0) = 0, f(1) = 1$).
Assume both machines start with the uniform superposition $|\psi\rangle = |0\rangle + |1\rangle$.

(a) **[5 pts]** For Machine A, compute $U_f|\psi\rangle$. Then, apply the filter to compute the final state

$H(U_f|\psi\rangle)$.

(b) **[5 pts]** For Machine C, compute $U_f|\psi\rangle$. Then, apply the filter to compute the final state $H(U_f|\psi\rangle)$.

(c) **[5 pts]** Use your results from parts (a) and (b) to determine how the final state $H(U_f|\psi\rangle)$ distinguishes constant machines from balanced machines. Prove that the final state is supported only on one basis state in the constant case and only on the other basis state in the balanced case.

PROBLEM 13: **20 Points**

Let's prove this rule holds for *any* 1-bit machine. Let $f : \{0,1\} \to \{0,1\}$ be promised to be either constant or balanced. Start with $|\psi\rangle = |0\rangle + |1\rangle$.

(a) **[5 pts]** Suppose f is constant. Prove that $U_f|\psi\rangle$ is either $(|0\rangle + |1\rangle)$ or $-(|0\rangle + |1\rangle)$.

(b) **[5 pts]** Use part (a) to show that if f is constant, measuring $H(U_f|\psi\rangle)$ will always result in a reading of $|0\rangle$.

(c) **[5 pts]** Suppose f is balanced. Prove that $U_f|\psi\rangle$ is either $(|0\rangle - |1\rangle)$ or $-(|0\rangle - |1\rangle)$.

(d) **[5 pts]** Use part (c) to prove that if f is balanced, measuring $H(U_f|\psi\rangle)$ will always result in a reading of $|1\rangle$, allowing you to classify the machine with 100% certainty in a single query.

PROBLEM 14: **15 Points**

Definition: The dot product of two vectors $v = (a, b)$ and $w = (c, d)$ is defined as $ac + bd$. If the dot product is exactly 0, the two vectors are perfectly perpendicular, which is called being "orthogonal."

The transformation matrix $H = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ has rows $r_1 = (1, 1)$ and $r_2 = (1, -1)$.

(a) **[5 pts]** Compute the dot product $r_1 \cdot r_2$.

(b) **[10 pts]** In quantum mechanics, states that can be reliably distinguished from one another must be orthogonal. Explain how the orthogonality of the H matrix allows the algorithm to physically separate the "constant" universe from the "balanced" universe without them overlapping.

PROBLEM 15: **20 Points**

Classical logic gates often destroy information. For example, if a classical AND gate outputs a 0, it is impossible to reverse the process to know for sure whether the original inputs were 00, 01, or 10. The information is permanently erased.

Quantum mechanics, however, requires perfect **reversibility**. Operators cannot erase information; they can only rearrange it. This means every quantum filter must act as its own "undo" button. Let's prove that the interference filter H is reversible.

(a) **[5 pts]** Apply the filter twice to the zero state. First, recall that $H|0\rangle = |0\rangle + |1\rangle$. Now, compute the expanded state of $H(H|0\rangle)$ by evaluating $H(|0\rangle + |1\rangle)$.

(b) **[5 pts]** Apply the filter twice to the one state. Compute the expanded state of $H(H|1\rangle)$ by evaluating $H(|0\rangle - |1\rangle)$.

(c) **[5 pts]** In standard quantum mechanics (which uses fractions to keep probabilities equal to 100%), applying the filter twice returns the exact original state: $H(H|\psi\rangle) = |\psi\rangle$. Because we are using an un-normalized model for this round to keep the math clean, applying H twice returns a scalar multiple of the original state. Based on your answers to (a) and (b), compute this scalar multiple.

(d) **[5 pts]** Show that the filter doesn't "destroy" any data and has perfect reversibility.

PROBLEM 16: **20 Points**

Now let the hidden machine take n inputs: $f : \{0,1\}^n \to \{0,1\}$. After querying the machine simultaneously using the uniform superposition, the state becomes:

$$U_f|\psi\rangle = \sum_{x\in\{0,1\}^n} (-1)^{f(x)}|x\rangle.$$

Recall from Part II (Problem 7) that you proved the amplitude sum $S = \sum_{x\in\{0,1\}^n}(-1)^{f(x)}$ must be exactly $\pm 2^n$ if f is constant, and exactly 0 if f is balanced.

(a) **[10 pts]** Let

$$S = \sum_{x\in\{0,1\}^n} (-1)^{f(x)}.$$

Prove that S is equal to the number of inputs x for which $f(x) = 0$ minus the number of inputs x for which $f(x) = 1$.

(b) **[10 pts]** Suppose the final interference filter sends the coefficient of $|00\cdots0\rangle$ to

$$S = \sum_{x\in\{0,1\}^n} (-1)^{f(x)}.$$

Prove whether the final measurement can output $|00\cdots0\rangle$ when f is balanced. Then prove whether the final measurement must output $|00\cdots0\rangle$ when f is constant.

Part IV: The Secret String

Note: This section combines the classical bounds of Part I, the phase mapping of Part II, and the interference filter from Part III to solve a fundamental limit of computer science.

There is a strict law in classical information theory: **To learn n independent bits of information, you must ask at least n yes-or-no questions.** If you want to know a 10-digit phone number, one single question will never be enough.

Your team has unlocked a deeper, more complex level of the hardware that tests this physical limit. This new machine is guarding a **Secret Password**—a specific, hidden binary string of length n, which we will call s.

When you input a string x, the machine computes the **bitwise dot product** of x and s, modulo 2.

$$f(x) = (x \cdot s) \pmod 2 = (x_1 s_1 + x_2 s_2 + \cdots + x_n s_n) \pmod 2$$

Example: Modulo-2 Dot Product

Suppose the hidden password is $s = 101$.

- If you input $x = 110$, the machine calculates $1(1) + 1(0) + 0(1) = 1 \pmod 2$. It outputs 1.
- If you input $x = 111$, the machine calculates $1(1) + 1(0) + 1(1) = 2 \pmod 2$. It outputs 0.

PROBLEM 17: **15 Points**

Suppose the machine takes 3-bit inputs ($n = 3$). A junior researcher queried the machine classically three times, but the computer crashed before they could finish testing all 8 inputs. They hand you the following data log:

Input (x)	Output $f(x)$
100	1
010	0
001	1

(a) **[5 pts]** Compute the secret string s.

(b) **[5 pts]** Let e_i denote the n-bit string with a 1 in the ith position and 0s elsewhere. Show that entering the following queries determines s

$$e_1, e_2, \ldots, e_n$$

(c) **[5 pts]** For a general n-bit machine, find the absolute minimum number of classical queries required to find the secret string s with 100% certainty.

PROBLEM 18: **15 Points**

Your team connects the new machine to the prototype quantum chip. You start with the uniform superposition $|\psi\rangle$ of all possible inputs. Just like in Part II, the machine encodes its output into the phase of the states:

$$U_f|\psi\rangle = \sum_{x \in \{0,1\}^n} (-1)^{x \cdot s} |x\rangle$$

Let $n = 2$ and assume the secret string is $s = 11$.

(a) **[5 pts]** Compute the dot product $(x \cdot s) \pmod 2$ for all four possible inputs: $x \in \{00, 01, 10, 11\}$.

(b) **[5 pts]** Determine the fully expanded quantum state $U_f|\psi\rangle$ for this specific machine, using $+/-$ signs.

(c) **[5 pts]** Looking at your resulting state, show why immediately measuring it right now is useless for finding s.

PROBLEM 19:

25 Points

To extract s, we must apply the interference filter from Part III to every single bit simultaneously. We denote this massive filter as $H^{\otimes n}$.

When applied to our n-bit state, the filter mathematically transforms it into the following double-sum (you do not need to derive this):

$$H^{\otimes n}(U_f|\psi\rangle) = \sum_{y\in\{0,1\}^n} \left[\sum_{x\in\{0,1\}^n} (-1)^{(x\cdot s)+(x\cdot y)} \right] |y\rangle$$

This looks intimidating, but it behaves exactly like the constructive and destructive collisions you explored in Part III. Let's evaluate the expression inside the brackets for a specific resulting state $|y\rangle$.

(a) **[5 pts]** **Constructive Interference:** Suppose we are looking at the specific state where $y = s$. The expression inside the brackets becomes $(-1)^{(x\cdot s)+(x\cdot s)}$. Prove that for any binary strings x and s, the value of $(x \cdot s) + (x \cdot s)$ is always even.

(b) **[5 pts]** Determine what the term $(-1)^{(x\cdot s)+(x\cdot s)}$ simplifies to.

(c) **[5 pts]** Using your answer to (b), compute the sum inside the brackets for $y = s$:

$$\sum_{x\in\{0,1\}^n} (-1)^{(x\cdot s)+(x\cdot s)}.$$

(d) **[5 pts]** **Destructive Interference:** It is a mathematical fact that if $y \neq s$, the positive and negative signs perfectly balance out, meaning the sum inside the brackets is exactly 0. Using this fact and your answer to (c), determine the final, un-normalized quantum state of the system.

(e) **[5 pts]** Based on your final state, find the number of quantum queries required to find the secret password s, regardless of how large n is.

PROBLEM 20:

25 Points

Your research team must draft a budget proposal to justify using the prototype quantum chip over standard classical server farms.

According to the hardware department:

- A **Classical Query** costs \$5 in electricity and takes 2 milliseconds.
- A single **Quantum Query** requires supercooling the hardware. It costs \$15,000 in liquid helium and takes exactly 300 milliseconds.

Assume you are trying to crack a secret string of length n.

(a) **[5 pts]** Find the function $T_c(n)$ for the time (in milliseconds) it takes the classical computer to find the string. Find the function $T_q(n)$ for the time it takes the quantum computer.

(b) **[5 pts]** Compute the minimum string length n does the quantum computer become *faster* than the classical computer?

(c) **[5 pts]** Find the function $C_c(n)$ for the cost (in dollars) of the classical computer. Find the function $C_q(n)$ for the cost of the quantum computer. Compute the minimum string length n that the quantum computer become *cheaper* than the classical computer?

(d) **[10 pts]** Modern bank encryption uses 2048-bit strings ($n = 2048$). If your team is tasked with cracking a bank's security, determine exactly how much time (in milliseconds) and how much money (in dollars) the quantum chip will save compared to the classical server farm.
