(1) (Bernstein-Vazirani problem)
For \( n \)-bit strings \( x = x_1 \ldots x_n \) and \( a = a_1 \ldots a_n \) in \( B_n \) we have the sum \( x \oplus a \) which is an \( n \)-bit string, and now introduce the 1-bit “dot product” \( x \cdot a = x_1 a_1 + x_2 a_2 + \ldots + x_n a_n \).
For any fixed \( n \)-bit string \( a = a_1 \ldots a_n \) consider the function \( f_a : B_n \rightarrow B_1 \) given by
\[
 f_a(x_1, \ldots, x_n) = x \cdot a
\]  
\((1)\) (Bernstein-Vazirani problem)

(a) Show that for any \( a \neq 00 \ldots 0 \), \( f_a \) is a balanced function i.e. \( f_a \) has value 0 (respectively 1) on exactly half of its inputs \( x \).
(b) Given a classical black box that computes \( f_a \) describe a classical deterministic algorithm that will identify the string \( a = a_1 \ldots a_n \) on which \( f_a \) is based. Show that any such black box classical algorithm must have query complexity at least \( n \).

Now for any \( n \) let \( H_n = H \otimes \ldots \otimes H \) be the application of \( H \) to each qubit of a row of \( n \) qubits. Show that (for \( x \in B_1 \) and \( a \in B_n \))
\[
 H \vert x \rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in B_n} (-1)^{xy} \vert y \rangle \quad H_n \vert a \rangle = \frac{1}{\sqrt{2^n}} \sum_{y \in B_n} (-1)^{ay} \vert y \rangle
\]
\((c)\) (the Bernstein–Vazirani problem/algorithm)

For each \( a \) consider the function \( f_a \) which is a balanced function if \( a \neq 00 \ldots 0 \) (as shown above). Show that the Deutsch-Jozsa algorithm will perfectly distinguish and identify the \( 2^n \) balanced functions \( f_a \) (for \( a \neq 00 \ldots 0 \)) with only one query to the function (quantum oracle for \( f \)). Indeed, show that the \( n \) bit output of the final measurements of the algorithm gives the string \( a \) with certainty for these special balanced functions.

(2) (Classical complexity – integer exponentiation mod \( N \))
Exponentiation of integers mod \( N \) is a basic arithmetic task (it will be used for example in Shor’s algorithm) and it is important to know that it can be done in \( \text{poly}(n) \) time where \( n = \log N \) is the number of digits for integers in \( \mathbb{Z}_N \).
To compute say \( 3^k \mod N \) (for \( k \in \mathbb{Z}_N \) and \( N > 3 \)) we could multiply 3 together \( k \) times. Show that this is not a \( \text{poly}(n) \) time computation. Devise an algorithm that does run in \( \text{poly}(n) \) time. (Hint: consider repeated squaring).
You may assume that multiplication of integers in \( \mathbb{Z}_N \) may be done in \( O(n^2) \) time. Generalise to a poly time computation of \( k_1^{k_2} \mod N \) for \( k_1, k_2 \in \mathbb{Z}_N \) showing that it may be computed in \( O(n^3) \) time.

(3) (Simon’s algorithm)
Simon’s decision problem is the following:
\textbf{Input:} an oracle for a function \( f : B_n \rightarrow B_n \),
\textbf{Promise:} \( f \) is either (a) a one-to-one function or (b) a two-to-one function of the following special form – there is a \( \xi \in B_n \) such that \( f(x) = f(y) \) iff \( y = x \oplus \xi \) (i.e. \( \xi \) is the period of \( f \) when its domain is viewed as being the group \((\mathbb{Z}_2)^n\)).
\textbf{Problem:} determine which of (a) or (b) applies (with any prescribed success probability \( 1 - \epsilon \) for any \( \epsilon > 0 \)).

It can be argued (e.g. as indicated in lecture notes) that for classical computation, this requires at least \( O(2n^{3/4}) \) queries to the oracle. In this question we will develop a quantum algorithm that solves the problem with quantum query complexity only \( O(n) \). Even more, the algorithm will determine the period \( \xi \) if (b) holds. Thus (unlike the balanced vs. constant problem) we
will have a provable exponential separation between classical and quantum query complexities, even in the presence of bounded error.

To begin, consider $2n$ qubits with the first (resp. last) $n$ comprising the input (resp. output) register for a quantum oracle $U_f$ computing $f$ i.e. $U_f |x⟩|y⟩ = |x⟩|y ⊕ f(x)⟩$ for $n$-bit strings $x$ and $y$.

(a) With all qubits starting in state $|0⟩$ apply $H$ to each qubit of the input register, query $U_f$ and then measure the output register (all measurements being in the computational basis). Write down the generic form of the $n$-qubit state $|α⟩$ of the input register, obtained after the measurement. Suppose we then measure $|α⟩$. Would the result provide any information about the period $ξ$?

(b) Having obtained $|α⟩$ as in (a), apply $H$ to each qubit to obtain a state denoted $|β⟩$. Show that if we measure $|β⟩$ then the $n$-bit outcome is a uniformly random $n$-bit string $y$ satisfying $ξ · y = 0$ (so any such $y$ is obtained with probability $1/2^{n−1}$).

Now we can run this algorithm repeatedly, each time independently obtaining another string $y$ satisfying $ξ · y = 0$. Recall that $B_n = (Z_2)^n$ is a vector space over the field $Z_2$. If $y_1, ..., y_s$ are $s$ linearly independent vectors (bit strings) then their linear span contains $2^s$ of the $2^n$ vectors in $B_n$. Furthermore to solve systems of linear equations over $B_n$ we can use the standard Gaussian elimination method (calculating with the algebra of the field $Z_2$), which runs in $\text{poly}(n)$ time.

(c) Show that if $(n−1)$ bit strings $y$ are chosen uniformly randomly and independently satisfying $y · ξ = 0$ then they will be linearly independent (and not include the all-zero string $00...0$) with probability

$$\prod_{k=1}^{n−1} \left(1 - \frac{2^{k−1}}{2^{n−1}}\right) = \frac{1}{2} \prod_{k=1}^{n−2} \left(1 - \frac{2^{k−1}}{2^{n−1}}\right).$$

Show that this is at least $1/4$. (It may be helpful here to recall that for $a$ and $b$ in $[0, 1]$ we have $(1−a)(1−b) ≥ 1 − (a + b)$).

(d) Show how the above may be used to solve Simon’s problem with $O(n)$ quantum query complexity (for any desired success probability $0 < 1 − ε < 1$).

(4) (Another query complexity problem with quantum advantage)

Let $B_n$ denote the set of all $n$-bit strings. The Hamming distance between two $n$-bit strings $a = a_1...a_n$ and $x = x_1...x_n$ is the number of places $j$ where $a_j$ and $x_j$ differ. Let $H_a : B_n → B_2$ be the function

$$H_a(x) = \text{(Hamming distance between a and x) mod 4}.$$ 

Here we are identifying $B_2$ with $Z_4$ via the usual binary representations of 0,1,2,3. (For example if $a = 101110000$ and $x = 001001110$ then $H_a(x) = 6 \text{ mod } 4 = 2$.)

Now consider the promise problem **HAM-mod4**: 

**Input**: a black box for a function $f : B_n → B_2$. 

**Promise**: $f$ is $H_a$ for some $n$-bit string $a$. 

**Problem**: determine $a$ with certainty.

In the quantum context the black box is a unitary operation on $(n + 2)$ qubits given by

$$U_f |x⟩|y⟩ = |x⟩|y + f(x)⟩.$$ 

Here the $x$ register is $n$ qubits and in the $y$ register we will write the basis as $\{|0⟩, |1⟩, |2⟩, |3⟩\}$ with addition in the expression $y + f(x)$ being addition in $Z_4$.

(a) Show that classically the query complexity of **HAM-mod4** is at least $n/2$. 


We will now show that the problem can be solved quantum mechanically with just one query. Let $M$ be the matrix

$$M = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}.$$ 

Note that $M$ is unitary. Also introduce the 1-bit functions $h_0, h_1 : B_1 \to B_1$ where

$$h_0(0) = 0 \quad h_0(1) = 1 \quad h_1(0) = 1 \quad h_1(1) = 0$$

i.e. $h_a$ is just $H_a$ for a 1-bit string $a$.

(b) For $a_1 = 0, 1$ show that

$$M \ket{a_1} = \frac{1}{\sqrt{2}} \sum_{x_1 = 0}^{1} i^{h_a(x_1)} \ket{x_1}.$$ 

(c) Returning to the case of $n$-bit strings $a = a_1 \ldots a_n$ and $x = x_1 \ldots x_n$ show that

$$H_a(x) = h_{a_1}(x_1) + \ldots + h_{a_n}(x_n) \mod 4.$$ 

Hence describe how the state

$$\ket{H_a} = \frac{1}{\sqrt{2^n}} \sum_{x \in B_n} i^{H_a(x)} \ket{x}$$

may be manufactured from $\ket{a}$.

(d) Let $S$ denote the 2-qubit “shift” operation

$$S \ket{y} = \ket{y + 1 \mod 4} \quad y \in \mathbb{Z}_4.$$ 

Let $QFT$ denote the quantum Fourier transform mod 4. Calculate the state $\ket{\psi_3} = QFT \ket{3}$ and show that $S \ket{\psi_3} = i \ket{\psi_3}$.

(e) Use the above results to show how $\text{HAM-mod4}$ may be solved with certainty using just one query to the oracle $U_f$ and poly($n$) total time complexity. (It may be helpful to note that $U_{H_a} \ket{x} \ket{y} = \ket{x} S^{H_a(x)} \ket{y}$.)

Draw a circuit diagram for your quantum algorithm.

[Optional afterthought: note that this algorithm is structurally “the same as” the Bernstein-Vazirani (BV) algorithm and it is interesting to compare the corresponding ingredients and their functionality. What are the BV ingredients corresponding to the use of $QFT$, $\ket{\psi_3}$, $M$, $h_a$ and $H_a$ here?] 

(5) (Approximately universal quantum gate sets)

(a) For unitary gates $U_1, V_1, U_2, V_2$ show that:

if $||U_1 - V_1|| \leq \epsilon_1$ and $||U_2 - V_2|| \leq \epsilon_2$ (i.e. the $V$’s are “approximate versions” of the $U$’s) then $||U_2U_1 - V_2V_1|| \leq \epsilon_1 + \epsilon_2$ i.e. “errors” in using approximate versions at most add when gates are composed.

(Recall that here $||U - V||$ is defined as the maximum length of the vector $(U - V) \ket{\psi}$ over all choices of normalised $\ket{\psi}$’s.)

Deduce that if $||U_i - V_i|| \leq \epsilon$ for $i = 1, \ldots, n$ then $||U_n \ldots U_1 - V_n \ldots V_1|| \leq n\epsilon$.

(b) For the purposes of this question you may assume the following: if a gate set $S$ is approximately universal then any one- or two-qubit gate $U$ may be approximated to within $\epsilon$ by a circuit of gates from $S$ of size $\text{poly}(1/\epsilon)$. (Actually by the Solovay-Kitaev theorem, mentioned in lectures, a stronger result is true viz. that a circuit of much smaller size $\text{poly}(\log(1/\epsilon))$ suffices, but we will not need that improvement here.)
Let $\mathcal{G}$ and $\mathcal{H}$ be two approximately universal sets of gates comprising one- and two-qubit gates only. Suppose that the decision problem $D$ is in the complexity class $\text{BQP}$ with all quantum gates in the circuits being from the set $\mathcal{G}$. Show that $D$ is then also in the class $\text{BQP}$ defined using quantum gates from the set $\mathcal{H}$ i.e. the definition of $\text{BQP}$ is independent of the choice of approximately universal set of gates used.

(6) (Period finding algorithm)
Consider the function $f(x) = 5^x \mod 39$ on the domain $x \in \mathbb{Z}_{2^n}$ with say $m = 11$ (as in fact would occur in Shor’s algorithm for factoring 39).
(a) Show that $f$ is periodic and determine its period $r$ (use a calculator.)
(b) Suppose we construct the equal superposition state $|f\rangle$ of $(x, f(x))$ values over the domain $\mathbb{Z}_{2^m}$, measure the second register, perform the quantum Fourier transform mod $2^m$ on the post-measurement state of the first register, and finally measure it. What is the probability for each possible outcome $0 \leq c < 2^m$ in the latter measurement? (Note: this should require very little calculation!) What is the probability that we successfully determine $r$ from this measurement result, using the standard process of the quantum period finding algorithm?

(7) (Entanglement is necessary for advantage in quantum computation)
Consider a quantum computation, given as a poly-sized circuit family $\{C_1, C_2, \ldots, C_n, \ldots\}$ where each $C_n$ comprises gates from a universal set $\mathcal{G}$ comprising one- and two-qubit gates, and suppose that this computation solves a decision problem $A$ in $\text{BQP}$.
Suppose further that for any input $x \in B_n$ to the circuit $C_n$ (for any $n$), at every stage of the process, the quantum state is unentangled i.e. it is a product state of all the qubits involved. Show that then the problem $A$ is also in $\text{BPP}$ i.e. if no entanglement is ever present in a quantum computation, then it cannot provide any computational benefit over classical computation (up to at most a polynomial overhead in time). (Hint: consider calculating the progress of the quantum process itself on a classical computer).