Problem set 6

CSE 534 Autumn 2026

Due date: November 16th 11:59 pm

Instructions: Solutions should be legibly handwritten or typeset. Mathematically rigorous solutions are expected for all problems unless explicitly stated. You are encouraged to collaborate on problems in small teams; however, each team member must write and submit their own individual solution. Read the AI tools policy on the course webpage and adhere to it. For grading purposes, start each problem on a new page.

Problem 1 (Understanding Grover's)**.**

1. **(2 points)** Give a circuit for running Grover's search on 2 qubits where $f(x) = 1$ for exactly one $x \in \{0,1\}^2$. What is the optimal number of Grover iterations needed, and what is the probability of finding x? Here, a Grover iteration is defined as applying both reflections once.

2. **(3 points)** Consider an f where there is a unique solution $x^\star$ to $f(x) = 1$ for $x \in \{0,1\}^n$. What happens if someone impatiently runs Grover's search by measuring the state of the quantum algorithm between each iteration of Grover's search to see if it has found a solution yet? Formally, between each Grover iteration, they measure according to the following projective POVM

$$\left\{ M_0 = \sum_{x\,:\,f(x)=0} |x\rangle\langle x|, M_1 = |x^\star\rangle\langle x^\star| \right\}, \text{ with post-measurement transforms given by } \rho \mapsto \frac{M_b \rho M_b}{\mathrm{tr}(M_b \rho)}.$$

   What is the expected number of Grover iterations till the measurement $\{M_0, M_1\}$ outputs $x^\star$?

3. **(2 points)** In class, we discussed, but did not prove, that Grover's search for finding some solution $x \in \{0,1\}^n$ when promised there are K solutions runs in time $\sqrt{2^n/K}$. Prove this statement. (No need to repeat the proof of Grover's from class. Just note the few lines where it will differ.)

4. **(2 points)** Consider running your modified Grover's search algorithm for K solutions when there are actually K' solutions. What is the probability that it will output a solution?

5. **(2 points)** Come up with a Grover's search algorithm for finding a solution $x \in \{0,1\}^n$ when the number of solutions is unknown. What is the runtime? If your algorithm is not deterministic, express the runtime in terms of N and δ, the probability of not outputting a solution.

Problem 2 (Quantum query lower bounds)**.** In class, we saw a proof of the famous Bennett, Bernstein, Brassard, and Vazirani (BBBV) 1994 result that showed that any quantum query algorithm that succeeds in solving unconstrained search over a set of size N requires $\Omega(\sqrt{N})$ queries.

Today, we are going to prove a different query lower bound, this time for the PARITY function. Let $\mathbf{z} : [N] \to \{0,1\}$ be any function. We define PARITY($\mathbf{z}$) to be the parity of the truth table of $\mathbf{z}$ — i.e., whether

the number of entries equal to 1 in the truth table of $\mathbf{z}$ is even or odd. A classical query algorithm for computing PARITY($\mathbf{z}$) requires N queries to $\mathbf{z}$ as the parity depends on every possible entry of the truth table. We are going to prove that any quantum query algorithm for computing PARITY($\mathbf{z}$) with probability $\geq 2/3$ for all inputs of $\mathbf{z}$ requires $\geq N/2$ queries.

1. **Step 1: Creating a polynomial**

   We can equivalently describe $\mathbf{z}$ by its truth table

   $$\text{truth table of } \mathbf{z} = \boxed{z_1}\boxed{z_2}\boxed{z_3}\boxed{z_4}\boxed{\cdots}\boxed{z_N}$$

   where each $z_x \in \{0, 1\}$. Then,

   $$\text{PARITY}(\mathbf{z}) = \bigoplus_{x \in [N]} z_x.$$

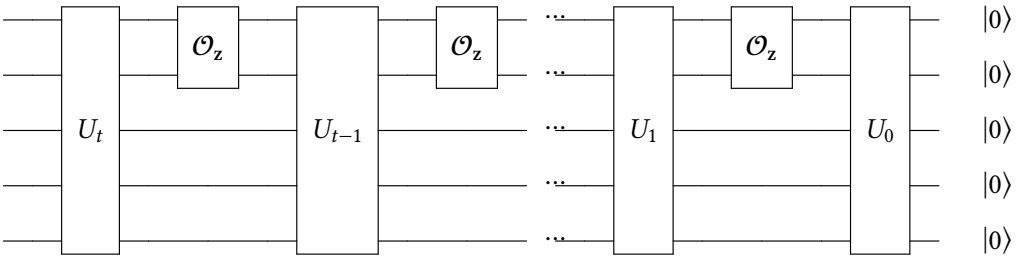
   Recall that a query to $\mathbf{z}$ (after conjugation by Hadmaard matrices) acts as

   $$\mathcal{O}_\mathbf{z} |x, b\rangle = (-1)^{b \cdot z_x} |x, b\rangle = (1 - 2b \cdot z_x) |x, b\rangle .$$

   Furthermore, it follows that

   $$\mathcal{O}_\mathbf{z} \otimes \mathbb{I} |x, b, y\rangle = (1 - 2b \cdot z_x) |x, b, y\rangle .$$

   (a) **(4 points)** Consider a quantum circuit that makes t queries to $\mathbf{z}$, interspersed with arbitrary unitary operations independent of $\mathbf{z}$ and no measurements (recall that by the principle of deferred measurement, we can remove measurements). Assume that it is an algorithm on m qubits and its initial state is $|0^m\rangle$.

   Write its output state as

   $$\sum_{s \in \{0,1\}^m} \alpha_s(z_1, \ldots, z_N) |s\rangle ,$$

where s ranges over the computational basis of all the algorithm's registers, including any workspace.

Prove that each amplitude α_s can be expressed as a *multilinear* polynomial with *complex* coefficients of total degree at most t in $z_1, \ldots, z_N$.

Hint: Induct on the number of queries. How does an oracle query affect the degrees of the amplitudes? How does a unitary independent of $\mathbf{z}$ affect them?

(b) **(1 point)** Argue that the probability of obtaining outcome 1 when measuring the first output qubit in the computational basis agrees, on Boolean inputs, with a multilinear polynomial $p(z_1, \ldots, z_N)$ with real coefficients and total degree at most $2t$.

2. **Step 2: Symmetrizing the polynomial**

(a) **(2 points)** For $0 \leq k \leq N$, let $j_1, \ldots, j_k$ be distinct indices of $[N]$. And for $i \in \{0, 1, \ldots, N\}$, let $S_i \subseteq \{0,1\}^N$ be the set of Boolean strings containing exactly i ones. Prove that

$$\frac{1}{|S_i|} \sum_{\mathbf{z} \in S_i} z_{j_1} \ldots z_{j_k} = \frac{(N-k)!}{N!} \cdot i(i-1)(i-2)\ldots(i-k+1). \tag{$\dagger$}$$

(b) **(2 points)** Using the multilinear polynomial p previously calculated, define for $i \in \{0, 1, \ldots, N\}$

$$q(i) \stackrel{\text{def}}{=} \frac{1}{|S_i|} \sum_{\mathbf{z} \in S_i} p(\mathbf{z}).$$

Prove that these values agree with a univariate polynomial $q : \mathbb{R} \to \mathbb{R}$ of degree at most $2t$.

Hint: Write the polynomial p as a linear combination of monomials. Each monomial will be of the form $z_{j_1} \cdots z_{j_k}$ and use eq. ($\dagger$) to generate a univariate polynomial in i of degree $\leq 2t$ for each monomial.

3. **Step 3: Shape of the polynomial**

(a) **(2 points)** Now assume that our quantum circuit with t queries actually correctly calculates PARITY($\mathbf{z}$) on all inputs $\mathbf{z}$ with probability $\geq 2/3$. What can we say about the value of $q(i)$ when i is even? What about when it is odd?

Draw a cartoon of what the polynomial q looks like.

(b) **(1 point)** Argue that the polynomial q has degree $\geq N$.

(c) **(1 point)** Conclude that $t \geq N/2$.