CS-151 Quantum Computer Science: Problem Set 5

Professor: Saeed Mehraban
TA: Dale Jacobs

Spring, 2024

Guidelines: ***The deadline to return this problem set is 11.59pm on Wednesday, March 6***. *Remember that you can collaborate with each other in the preliminary stages of your progress, but each of you must write their solutions independently. Submission of the problem set should be via Gradescope only. Best wishes!*

Problem 1 (25 points). *Recall in class we discussed two reversible (quantum) query models for an arbitrary Boolean function $f : \{0,1\}^n \to \{0,1\}$. These are,*

- *The phase-query model, defined by $P_f : |x\rangle \mapsto (-1)^{f(x)} |x\rangle$.*
- *The index-query model, defined by $I_f : |x\rangle \otimes |a\rangle \mapsto |x\rangle \otimes |a \oplus f(x)\rangle$, where*

a) Prove that you can simulate P_f using one query to I_f. (Hint: Don't create unnecessary superpositions. If you start with a state $|x\rangle \otimes |\psi\rangle$, the final result of your computation must be $(-1)^{f(x)} |x\rangle \otimes |\psi\rangle$.)

b) Prove that you can simulate I_f using one query to a controlled-P_f gate,

$$C(P_f) : |b\rangle \otimes |x\rangle \mapsto (-1)^{b \cdot f(x)} |b\rangle \otimes |x\rangle$$

(Hint: Think about the circuit you used for the Hadamard test.)

c) (Extra credit) Is it possible to do part (b) with a single phase-oracle that is not controlled on other bits? Give a circuit or prove that it is impossible.

Problem 2 (25 points). *Suppose you have query-access to a black-box function $f : \{0,1\}^n \to \{0,1\}$ and let s be the number of solutions to this function. Define $N := 2^n$.*

a) Start with n qubits initialized to the zero state, $|0 \cdots 0\rangle$ stored in the registers $1, \cdots, n$. Apply a Hadamard gate to each qubit. Now query f, and store the result in an ancilla $(n+1)^{th}$ register whose initial state was $|0\rangle$. Once again, apply a Hadamard gate to registers $1, \cdots, n$. What is the final state produced by this algorithm?

b) What is the probability that upon measuring the qubits $1, \cdots, n$, you get the all-zeroes state back?

c) Plot the probability from the previous step as a function of s/N where s and N are defined above. What's the minimum value of this probability for any $0 \leq s \leq N$?

Problem 3 (25 points). *In this problem, you will explore some applications of the Deutsch-Josza algorithm,*

a) *Let n be an even number and $x \in \{0,1\}^n$ be a length-n binary string. We define the PARITY of the string x as being 1 if the number of ones in x are odd, and 0 otherwise. If $x = (x_1, \cdots, x_n) \in \{0,1\}^n$, how many classical queries to x are required to calculate its parity? (Note: In this problem, you are allowed to 'query' the input string rather than a function. That is, the query $|0\rangle$ of x will return x_1, querying with $|1\rangle$ will return x_2, etc.)*

b) *Use Deutsch-Josza as a sub-routine to establish a quantum algorithm that uses $n/2$ queries to the string $x = (x_1, x_2, \cdots, x_n)$ to decide its PARITY with probability 1.*

c) *Assume that you are given query access to an input $x = (x_1, x_2)$ with $x_1, x_2 \in \{0,1\}^n$ with the promise that either $x_1 = 0^n$ and $x_2 = 1^n$, OR, the number of ones in x_1 plus the number of zeroes in x_2 is n. Modify the Deutsch-Josza algorithm to give a quantum algorithm that can decide which is the case. (Hint: You may want to calculate the probability of obtaining the all-zeros state at the end of your algorithm.)*

Problem 4 (25 points). *In this problem we will see another way of analyzing Simon's problem.*

a) *Let W be a subspace of $\{0,1\}^n$ and let*

$$|W\rangle = \frac{1}{\sqrt{|W|}} \sum_{w \in W} |w\rangle$$

be the n-qubit state that is in a uniform superposition over the elements of W. Let

$$W^{\perp} = \{v \mid v \cdot w = 0 \text{ for all } w \in W\}.$$

be the subspace orthogonal to W. Show that $H^{\otimes n}$ maps $|W\rangle$ to the n-qubit state $|W^{\perp}\rangle$.
(Note that $v \cdot w$ is the dot product over $\mathbb{F}_2^n$ as defined in class, $v \cdot w = v_1 \cdot w_1 \oplus \ldots \oplus v_n \cdot w_n$.)

b) *Recall in Simon's algorithm, we first obtain the state*

$$|\psi\rangle = \frac{1}{\sqrt{2^n}} \sum_x |x\rangle |f(x)\rangle$$

and then we measure the second half of the state to obtain

$$|\psi'\rangle = \frac{|x\rangle + |x \oplus s\rangle}{\sqrt{2}}$$

for some x. Using part (a), find an expression for $|\psi''\rangle = H^{\otimes n} |\psi'\rangle$ and show if we measure $|\psi''\rangle$ in the computational basis we always sample a string v that is orthogonal to s.

Simon's algorithm is completed by obtaining n different values satisfying $v_i \cdot s = 0$ and solving the corresponding system of n linear equations over the bits of v_i and s_i.

Problem 5 (Extra credit). *Imagine there are a number of students in a class, each having a random birthday. How many students do we need to have two students with the same birthday? We might think that we need exactly 365. However, a theorem known as the 'birthday paradox' says that with around 20 students, we will have a 'collision' between birth dates with high probability. In order to formalize the birthday paradox, suppose we have t items, each taking one of the values $\{1, \ldots, n\}$ uniformly at random. How many variables do we need in order to obtain the same number twice?Prove that there is a constant c_1 such that if $t > c_1\sqrt{n}$ then with high probability we get a collision. Furthermore, show that there is another constant $c_2 < c_1$ such that if $t < c_2\sqrt{n}$ then with high probability we won't have a collision.*

Recall in the setting of Simon's problem we have a Boolean function over n bits and a secret s such that for any $x, y \in \{0,1\}^n$, $f(x) = f(y)$ iff $y = x \oplus s$. In class, we told you that in order to solve this problem deterministically *we need at least 2^{n-1} queries in the worst case. Use the birthday paradox to show that the number of classical queries needed to solve Simon's problem with high probability is $\Theta(2^{n/2})$: show that there are two constants $c_2 < c_1$ such that if we query more than $c_1 2^{n/2}$ then we can solve this problem and if we query less than $c_2 2^{n/2}$ we can't.*