COS 598a
March 26, 2001
Prepared by Guest Lecturer: Yaoyun Shi

Assignment 4

Homeworks We use the notations in the Appendix.

Exercise 1 **Grover's algorithm is optimal.** There are many ways to prove that Grover's algorithm is optimal. Consider the following approach. Recall that the unordered search problem is to find i given the input $\overrightarrow{e}_i$, which is the binary string with the only '1' at the ith position. For any quantum decision tree algorithm for the unordered search problem, let $\phi_i^{(t)}$ be the state of the computer with input $\overrightarrow{e}_i$ and after U_t is applied, and P_i be the projection to the subspace $\mathbf{span}\{|i,b,c\rangle : \text{for all } b, c\}$. Prove that

$$\mathbf{E}_i \|P_i \phi_i^{(t)}\| = O(t/\sqrt{n}). \tag{1}$$

Exercise 2 **Element Distinctness.** The well known Element Distinctness problem is to decide whether or not the input numbers are all distinct. Prove that any comparison-based quantum algorithm for Element Distinctness must compare $\Omega(\sqrt{n}\log n)$ times, where n is the number of elements, and the type of comparison allowed is "$\geq$".

Hint: Modify the adversary for sorting in [17].

Exercise 3 **Claw-finding.** Given $m+n$ numbers $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$, the Claw-finding problem is to find one pair of indices i and j such that $x_i = y_j$.

- Assume $m = \sqrt{n}$. Design a quantum algorithm for Claw-finding that uses $O(\sqrt{n}\log n)$ comparisons.
- Assume $m = n$. Design a quantum algorithm for Claw-finding that uses $O(n^{3/4}\log n)$ comparisons.

Remark: By dividing the inputs randomly into two halves and then apply the second Claw-finding algorithm, Element Distinctness can be solved in $O(n^{3/4}\log n)$ comparisons.

Research Problems

Problem 1 One can easily prove that to find the medium any quantum algorithm needs to compare $\Omega(n)$ times. Assume that the input numbers $x_1, x_2, \ldots, x_n$ correspond to a permutation σ of $1, 2, \ldots, n$. Let P_σ be the projection to the subspace that corresponds to comparing the medium of σ with any other element. Prove or disprove that,

$$\mathbf{E}_\sigma \|P_\sigma \phi_\sigma^{(t)}\| = O(t/n). \tag{2}$$

Problem 2 Consider the first question in Exercise 3. One can easily prove a lower bound of $\Omega(\sqrt{n})$ comparisons. Can you close up the gap between this lower bound and the $O(\sqrt{n}\log n)$ upper bound? Does the technique in [17] work for gaining a *logn* factor in the lower bound?

Problem 3 Can you close up the gap between the best known upper bound ($O(n^{3/4}\log n)$) and lower bound ($\Omega(\sqrt{n}\log n)$) for Element Distinctness in the comparison model?

Notes The results on the topic discussed in class and in the Appendix are from Grover [16], Høyer and Neerbek and Shi [17]. For information on the topic discussed in Exercise 3 and Research Problem 2, see Buhrman et al. [15]. The quantum complexity of Element Distinctness has attracted a lot of efforts in recent years. Nevertheless, there is still a big gap between the best known upper and lower bounds.

Appendix: The quantum decision tree model

We give a formal definition of the standard quantum decision tree model for computing a function $f : \{0,1\}^m \to \{0,1\}^k$.

The quantum algorithm works in the Hilbert space

$$\mathbf{span}\{|s,b,c\rangle : s \in \mathbb{Z}_m,\ b \in \{0,1\},\ c \in \{0,1\}^w, \text{ for some integer } w \geq 0\}.$$

An oracle holds the input $x = x_0 x_1 \cdots x_{m-1} \in \{0,1\}^m$, which is not known to the algorithm directly. Recall that in the classical decision tree model, the input bit x_i is returned to the algorithm if the index i is presented to the oracle. In the quantum setting, due to the requirement of being unitary for all quantum operations, x_i is XOR-ed with another bit provided by the algorithm. Mathematically, the algorithm "reads" the input through the *oracle gate*, which is a unitary transformation that depends on x and works on each base vector $|s,b,c\rangle$ as follows:

$$O_x|s,b,c\rangle = |s, b \oplus x_s, c\rangle.$$

It is easy to verify that with $\Psi^{\pm} = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$,

$$O_x|s,\Psi^-,c\rangle = (-1)^{x_s}|s,\Psi^-,c\rangle,$$

and,

$$O_x|s,\Psi^+,c\rangle = |s,\Psi^+,c\rangle.$$

A quantum decision tree algorithm that makes T queries is the application of a sequence of unitary transformations on the initial state $|\overrightarrow{0}\rangle$:

$$U_T O_x U_{T-1} O_x \cdots U_1 O_x U_0 |\overrightarrow{0}\rangle.$$

For all t, $0 \leq t \leq T$, we write

$$\phi_x^{(t)} := U_t O_x \cdots U_0 |\overrightarrow{0}\rangle.$$

We say that the quantum algorithm computes f with the error probability bounded by ϵ, for some constant ϵ such that $0 \leq \epsilon < 1/2$, if there is a measurement M, such that for any input x, when M is applied to $\phi_x^{(T)}$, the probability of observing $f(x)$ is no less than $1 - \epsilon$.