**Lecture & Proseminar 250078/250042**
**"Quantum Information, Quantum Computation, and Quantum Algorithms" WS 2022/23**

**— Exercise Sheet #10 —**

**Problem 26: Grover's algorithm with multiple marked elements.**

Consider the Grover search problem of finding $x_0$ such that $f(x_0) = 1$ for a given function $f : \{0, N-1\} \to \{0, 1\}$. In the lecture, we derived Grover's algorithm which finds $x_0$ given that it is unique. In this problem, we will derive a generalization of Grover's algorithm which allows to tackle the search problem in the case where there are $K > 1$ solutions $x$ to the equation $f(x) = 1$. The goal is to find one $x$ with $f(x) = 1$ with high probability.
The oracle is constructed the same way as before, i.e., it acts as

$$O_f = \mathbb{I} - 2 \sum_{x:f(x)=1} |x\rangle\langle x| \ .$$

The algorithm proceeds the same way as before, namely, by starting in the state $|\omega\rangle$ (given in the lecture), repeatedly applying Grover iterations $G = -O_\omega O_f$ (with $O_\omega$ as in the lecture), and finally measuring in the computational basis.

1. Show that $O_f$ can be obtained from $U_f : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle$.
2. Show that the Grover iteration $G$ leaves the space $\mathcal{S} = \mathrm{span}(|\omega\rangle, |x_0\rangle)$ invariant, where $|\omega\rangle$ is as in the lecture, and
$$|x_0\rangle \propto \sum_{x:f(x)=1} |x\rangle \ .$$
3. What is the action of $G$ on a state in $\mathcal{S}$?
4. For a given number of solutions $K$, how many times do we have to apply $G$ to get a good overlap with $|x_0\rangle$? What result will we get when measuring in the computational basis?
5. Compare this to the scaling of the classical algorithm (i.e. trying random $x$ until a solution is found).

**Problem 27: Quantum counting.**

Consider the same setting and notation as in Problem 26. Here, we will use a combination of Grover iterations $G$ and phase estimation (Problem 25 on Sheet #9) to estimate ("count") the number $K$ of solutions up to some error $\delta K$. Our goal will be to understand how the accuracy $\delta K$ scales with the number $Q$ of queries to $f$ (or $U_f$).

1. First, determine the scaling $\delta K$ for classical counting: Since we assume that $f$ is a black-box function, the best we can classically do is to sample $Q$ random values $x_i$, $i = 1, \ldots, Q$, compute $f(x_i)$, and use this to estimate $K$. What is the error $\delta K$ as a function of $Q$ (and $K$, $N$)?
2. We will now construct a quantum algorithm for estimating $K$. First, determine the eigenvalues $e^{i\theta_k}$, $k = 1, 2$, of $G$ restricted to the subspace $\mathcal{S}$. (This is most easily done by observing that $G$ is a rotation by an angle $2\phi$ with $\sin\phi = \sqrt{K/N}$ – cf. Problem 26 – in this two-dimensional space.)
3. Now assume we are given one of the corresponding eigenvectors $|\theta_k\rangle$. We can now use the phase estimation algorithm to determine the phase $\theta_k/2\pi$ corresponding this eigenvector up to some number $d$ of digits. What is the number of queries to $O_f$ required for that? What is the resulting accuracy of $\theta_k$? (You can assume that the phase estimation is exact, i.e. neglect the additional error arising from the fact that $\theta_k/2\pi$ does not stop after $d$ digits.)

4. From $\theta_k$, we can estimate $K$. What is the error $\delta K$ as a function of $Q$ (and $K$, $N$)?

5. Show that this algorithm can be adapted to work also if we cannot prepare the state $|\theta_k\rangle$, but rather start in some other easy-to-prepare state in the subspace $\mathcal{S}$.

**Problem 28: Fast Fourier transform.**

In this problem, we will use the expression

$$\hat{\mathcal{F}} : |j_1, \ldots, j_n\rangle \mapsto \tfrac{1}{2^{n/2}} \big(|0\rangle + e^{2\pi i\, 0.j_n}|1\rangle\big) \otimes \big(|0\rangle + e^{2\pi i\, 0.j_{n-1}j_n}|1\rangle\big) \otimes \cdots \otimes \big(|0\rangle + e^{2\pi i\, 0.j_1 j_2 \ldots j_n}|1\rangle\big) \quad (1)$$

for the quantum Fourier transform $\hat{\mathcal{F}}$ derived in the lecture to construct an algorithm for the classical Fourier transformation on vectors of length $N = 2^n$ which scales as $O(2^n n) = O(N \log N)$ – the fast Fourier transformation (FFT) – as opposed to the naive $O(N^2)$ scaling.
Recall that the classical Fourier transformation $\mathcal{F} : \mathbb{C}^N \to \mathbb{C}^N$ acts as $\mathcal{F} : (x_0, \ldots, x_{N-1}) \mapsto (y_0, \ldots, y_{N-1})$, where

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} e^{2\pi i\, jk/N} x_j \ . \quad (2)$$

1. Show that performing the classical Fourier transformation by directly carrying out the sum in Eq. (2) requires $O(N^2)$ elementary operations.

2. As shown in the lecture, $\hat{\mathcal{F}}$ maps $\sum_j x_j|j\rangle$ to $\sum_k y_k|k\rangle$. Use this, combined with Eq. (1), to derive an explicit expression for $y_k$ in terms of the $x_j$ in the spirit of Eq. (1).

3. The resulting expression for $y_k$ as a function of the $x_j$ should contain a sum over $j_1, \ldots, j_n$. Show that this sum can be carried out bit by bit. (What should happen is that in each step, the "input" $x_j$ is transformed to a vector where one $j_i$ disappears due to the sum, and instead a dependency on one of the $k_\ell$ appears.)

4. What is the number of elementary operations required for each of these transformations? What is the total computational cost of the algorithm?