Introduction to Quantum Computation, UPB
Winter 2022, Assignment 9

Due: January 26, at start of tutorial

This homework gets you to practice using Grover's algorithm and amplitude amplification.

1 Exercises

1. Recall the unstructured search problem, for which we are given black-box access to the bits of a string $x = x_1 \cdots x_N \in \{0,1\}^N$, for $N = 2^n$. The goal is to find a solution, i.e. an index i such that $x_i = 1$, if such an index exists. Denote by t the Hamming weight x, i.e. the number of bits of x which are set to 1; note that t is unknown to you.

   In the lecture, we saw that running the Grover iterate $k = O(\sqrt{N/t})$ times suffices to find a solution. However, this depended on knowing t, which one does not know in general. Nevertheless, it is possible to find a solution even without knowing t using an expected number of $O(\sqrt{N/t})$ queries. In this question, we will see how.

   Consider the following algorithm:

   i. Set $m = 1$, $\lambda = 6/5$.
   ii. Choose j uniformly at random from set $\{1, \ldots, m\}$.
   iii. Apply the Grover iterate j times.
   iv. Measure, and stop if a solution is found.
   v. Else, set $m = \min(\lambda m, \sqrt{N})$. Return to Step ii.

   We now analyze this algorithm. Assume throughout this question that $t \leq 3N/4$. (Why is the problem trivial to solve if $t > 3N/4$?)

   (a) Set θ such that $\sin^2 \theta = t/N$, and $m_0 = 1/\sin(2\theta)$. Show that $m_0 < \sqrt{N/t}$.

   (b) It can be shown that whenever $m \geq 1/\sin(2\theta)$, the probability of obtaining a correct solution in step iv is at least $1/4$. Thus, intuitively, we wish to increment m up to at least $1/\sin(2\theta)$ in step v as quickly as possible, but without using more than $O(\sqrt{N/t})$ queries. This explains the use of a geometric progression obtained by multiplying by $\lambda > 1$ each time step v is run.

   What is the number of times step ii has to be run before we are guaranteed $m \geq 1/\sin(2\theta)$?

   (c) Once $m \geq 1/\sin(2\theta)$, we shall say the algorithm has reached the "critical stage". Show that the algorithm requires $O(\sqrt{N/t})$ queries to reach the critical stage.

   (d) Suppose the algorithm reaches the critical stage. Show that the expected number of Grover iterations needed to find a solution is now $O(\sqrt{N/t})$. Conclude that the total number of queries required by the algorithm is $O(\sqrt{N/t})$.

2. Assume we have black-box query access to a sequence of $N = 2^n$ distinct integers, $x = x_0 x_1 \cdots x_{N-1}$ for $x_i \in \mathbb{Z}$. Specifically, we assume the ability to apply both the query map $O_x : |i, 0\rangle \mapsto |i, x_i\rangle$ and its inverse. Note that each integer x_i can be arbitrarily large, i.e. do not assume each integer is at most (say) N in absolute value.

   Devise a quantum query algorithm, which with probability at least 2/3, finds the minimum x_i over all indices i. You should use $O(\sqrt{N})$ queries. You do not need to formally work out the full details of the algorithm and its analysis. Rather, describe the algorithm and argue why its runtime should be $O(\sqrt{N})$ at a high level.

   Hint: Try to do something along the lines of running Quicksort while being able to run Grover's algorithm as a subroutine. For your runtime analysis, you may simplify the Quicksort analysis by assuming that picking an index i uniformly at random yields "roughly" the median of the sequence x, i.e. approximately half the indices j will have $x_j < x_i$.