COMS 4281 - Introduction to Quantum Computing Fall 2025

Practice Worksheet 6 - Quantum algorithms for search and counting

This practice worksheet is intended to cover material up to November 11. The weekly quiz (due November 14th, 11:59pm) will be based on this worksheet. The final exam will have questions inspired by the worksheets.

Problem 1:

Given oracle access to a function $f : \{0,1\}^n \to \{0,1\}$, Grover's algorithm can find a marked input x such that $f(x) = 1$ using at most $O(\sqrt{2^n})$ queries. However, *query* complexity is not necessarily the same as *time* or *gate* complexity: if you were to implement Grover's algorithm on a quantum computer, there will be computational resources required to implement the actual oracle query.

(a) Suppose that there is a quantum circuit for computing O_f (the phase oracle corresponding to f) using T gates. What is the gate complexity of Grover's algorithm overall?

(b) Explain, in your own words, why the title of Lov Grover's original paper on the Grover search algorithm, "A fast quantum mechanical algorithm for database search," is a misleading indication of the use cases of Grover's algorithm?

(c) What kinds of functions f are better suited for using a quantum computer to solve the unstructured search problem?

Problem 2: Grover diffusion operator

Recall the Grover diffusion operator

$$R = 2\left|s\right\rangle\left\langle s\right| - \mathrm{I}$$

where $\left|s\right\rangle = \left|+\right\rangle^{\otimes n}$.

(a) What is $H^{\otimes n}\left|s\right\rangle$? What is $H^{\otimes n}\left|s\right\rangle\left\langle s\right| H^{\otimes n}$?

(b) Prove that R is the same thing as

$$R = H^{\otimes n} O_{\mathrm{OR}} H^{\otimes n}$$

where O_{OR} is the phase oracle corresponding to the n-bit OR function.

Hint: One way to prove this is to show that both the left-hand side and right-hand side act the same on all basis states.

(c) What is the gate complexity of implementing R on a quantum computer? Assume that you can apply a universal gate set consisting of single- and two-qubit gates.

Hint: Break down the n-bit OR function into a classical circuit consisting of AND, OR, NOT gates. Then convert it to a quantum circuit just like you did in Pset1.

Problem 3: Analyzing Grover's algorithm with multiple solutions

In class we focused on the case where the function f has a unique marked element x^* such that $f(x^*) = 1$. Let's think about the case where there are M solutions, out of domain size $N = 2^n$. Define

$$|\Delta\rangle = \frac{1}{\sqrt{N-M}} \sum_{x:f(x)=0} |x\rangle \qquad \text{and} \qquad |\Gamma\rangle = \frac{1}{\sqrt{M}} \sum_{x:f(x)=1} |x\rangle \ .$$

(a) Show that the state of Grover's algorithm before *any* Grover iterations is

$$|\psi_0\rangle = \cos(\theta)\,|\Delta\rangle + \sin(\theta)\,|\Gamma\rangle$$

where $\theta = \sin^{-1}(\sqrt{M/N})$.

(b) Show that the state after *one* Grover iteration is

$$|\psi_1\rangle = \cos(3\theta)\,|\Delta\rangle + \sin(3\theta)\,|\Gamma\rangle \ .$$

(c) Show that after k iterations the state is

$$|\psi_k\rangle = \cos((2k+1)\theta)\,|\Delta\rangle + \sin((2k+1)\theta)\,|\Gamma\rangle \ .$$

(d) Suppose we measure the state $|\psi_k\rangle$ in the standard basis. What is the distribution of outcomes?

(e) How many iterations k should Grover's algorithm be run so that we get a solution with high probability? For simplicity, you can assume that $M \ll N$, so that the small angle approximation works $\sin(\theta) \approx \theta$.

Problem 4:

What happens if you don't know the number of solutions M?

(a) Running the algorithm for $k \approx \sqrt{N}$ iterations can actually be problematic. Give an example of a function f such that running for $k = \sqrt{N}$ iterations, and then measuring, actually yields a solution with very *low* probability.

(b) Suppose that, instead of choosing a *fixed* number of iterations, Grover's algorithm chooses a *random number* of iterations $1 \le k \le \pi\sqrt{N}/4$ to run before stopping and measuring to see if there is a solution.

The expected probability of getting a solution is thus

$$\frac{4}{\pi\sqrt{N}} \sum_{k=1}^{\pi\sqrt{N}/4} \sin^2((2k+1)\theta)$$

where $\theta = \sin^{-1}(\sqrt{M/N})$. Write some code (in Python or whatever your favorite language is – you can use AI for this part if that's helpful) to plot this expected probability for $N = 1000$, and M varying between 1 and 1000. What does this probability tend to be?

(c) Given the number above, describe a way to augment Grover's algorithm (which picks a random number of iterations to run) so that it finds a solution with high probability, with $O(\sqrt{N})$ queries to f, *no matter how many solutions M there are.*