CS333 - Programming Assignment

Due: Anytime before the last day of class

For this programming assignment, you will investigate how realistic errors affect quantum algorithms. Existing quantum computers currently have a lot of errors. Eventually, once the errors are below a critical threshold, we will be able to use error correction to fix any errors and perform arbitrarily long quantum computations. However, the current errors are above the threshold, and can't be efficiently corrected. This means that longer circuits will typically not output the correct result, because there is time for errors to accumulate and propogate.

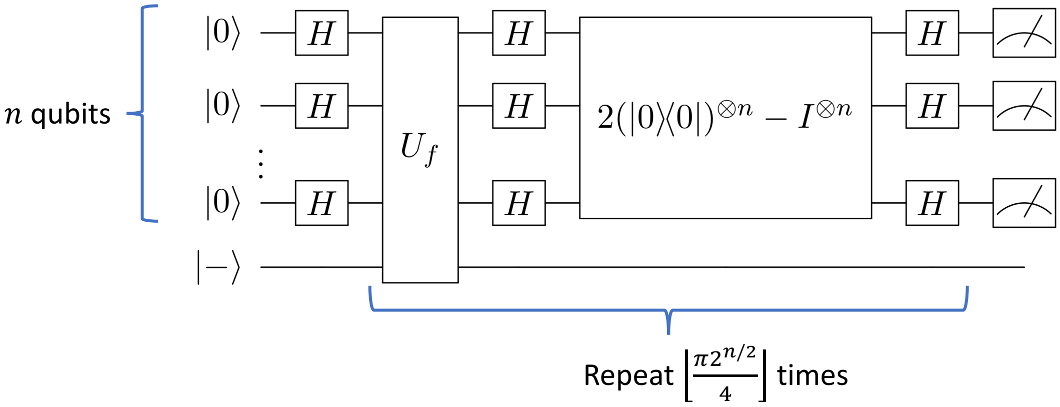
For this programming assignment, you should do the following:

1. Please write a program using pyQuil that takes as input a natural number n and a number s where $0 \leq s \leq 2^n - 1$, and outputs a quantum program (in Quil) for Grover search on n qubits where the marked element is s. (The circuit for Grover's search is below.)

2. Simulate your program using the Rigetti QVM, and show that it correctly finds s (with very high probability).

3. Simulate the computation using the noise simulator feature that simulates realistic noise on the Rigetti quantum computer. For small values of n, you should hopefully find that the algorithm is succesful at least some of the time. At larger n, you should find that noise overwhelms the computation, and the output is incorrect most of the time. Please plot the success probability as a function of n. At what value of n do you lose all quantum advantage? (That is, at what point is the quantum algorithm's success no better than guessing?)

The family of circuits for Grover's algorithm is as follows:


n qubits
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|-\rangle$
H
U_f
$2(|0\rangle\langle 0|)^{\otimes n} - I^{\otimes n}$
Repeat $\left\lfloor \frac{\pi 2^{n/2}}{4} \right\rfloor$ times

Here $f : \{0,1\}^n \to \{0,1\}$ is a function that takes value 1 on exactly on input, and is zero everywhere else. Let's call s the 1-valued input of f, so $f(s) = 1$, where $s \in \{0,1\}^n$. Then $U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle$ is an n-bit string if $|x\rangle$ and $|y\rangle$ are standard basis states. (Here $|x\rangle$ is a standard basis state of n qubits, and $|y\rangle$ is a standard basis state of a single qubit.) Generally one assumes that one can apply the unitary U_f, but s is unknown; the goal of the algorithm is to determine s.

Classically, it takes 2^n queries to f to figure out s, because s is one of 2^n possible inputs. However, if Grover's algorithm works correctly, the output should be a binary representation of s, which requires only $O(\sqrt{2^n})$ uses of U_f.

Please submit to Canvas:

- A python program that generates your QUIL program. This program should be appropriately commented.
- Sample input/output from the error-less simulation of your program for a couple of n's and s's, showing it works correctly.
- The code you used to collect data for the success of different sized algorithms under realistic noise. This should also be commented.
- A plot of success versus size. This should be clearly labelled.

Note: In Pyquil, if you have defined a multiqubit unitary that you label e.g. as `Uf`, and you want to have it act on qubits 3, 5, and 6, for example, you can write this instruction to a program `p` as `p.inst(("Uf"+" 3 5 6"))`

Note: I would expect that you take around 20 hours (or less) on this assignment. I will give partial (extra) credit based on how much you accomplish. Your grade will depend on clarity as well as content, so please comment and explain throughout. If you turn the assignment in early, I will try to give you feedback in case you'd like to turn in again.