CSCI 4140 Laboratory Three

Fundamental Algorithms, Part One

Introduction

This laboratory is an exploration of some of the fundamental algorithms. A chance to work with their implementations and explore some of their properties.

Quantum Key Distribution

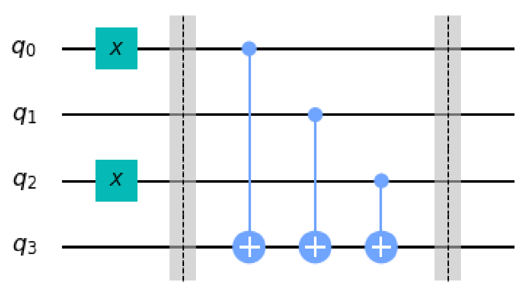
Python code for quantum key distribution is available in one of the video lectures and in the Qiskit textbook. Start by copying this code (yes, I said you could do that), and run the following two experiments.

First, change the length of the bit sequence that is generated and observe the change in efficiency. The length is the variable n, set on the second line of code. Try values of 100, 200, 500 and 1000 and record the efficiency. How long of a sequence do you expect you will need to get close to 50% efficiency? Record your results in your laboratory report in the form of a table.

Second, in the video lecture we observed that when the message was intercepted there was only a small number of bits that were different in the sample. Change the seed of the random number on the first line of code and observe how the number of different bits in the sample changes. Try five different seed values and record the number of different bits. Record your results in your laboratory report in the form of a table.

Balanced Functions

In the video lecture on the Deutsch-Josza algorithm a circuit was presented that claims to implement a balanced function:



There was no justification or explanation, it was just pulled out of thin air. In this part of the lab you will verify that this is indeed a balanced function. No math is required for this, just a little bit of programming. With three qubits as inputs there are 8 possible combinations of input values. To determine if this circuit is balanced, we only need to run it 8 times, once with each combination, and measure the result. The easiest way to do this is with a for loop that goes through the 8 possible value. To pick off the individual bits you can use the bitwise and operation (&) in Python. Create the circuit for the balanced function once and store it in a variable, this is what we did the video lecture, so you can copy that code. In the for loop create a new circuit with n+1 qubits and 1 classical bit in the following way:

- Call QuantumCircuit(n+1, 1)
- Set the first three qubits to the appropriate values using the & operator on the loop variable
- Add the balanced function circuit
- Measure from the last qubit to the classical bit

Once the circuit has been created simulate it with the qasm simulator. You can get the backend before the for loop. Use the following statement:

```
result = execute(qc, backend, shots=1, memory=True).result()
```

The last parameter saves the results of the computation so we can retrieve the value of the classical bit. The value of the classical qubit can be retrieved by calling result.get_memory()[0].

Is the function balanced? Copy your program code and the results and add them to your report.

Laboratory Report

Your laboratory report will have three sections. The first two sections report the results from the first part of the lab, and the third section has the results from the second part. Package this as a PDF or PNG file and submit it.