QOSF Mentorship Program: Cohort 4, Screening Task 1

Mike Nelson (mnelsonqcr@gmail.com)

September 22, 2021

Introduction: Formulating the problem

The goal of this task is to design a circuit that accepts a vector of the form

$$[1, 5, 7, 10],$$

and outputs a quantum state which is a linear superposition of the indices (in binary) of the target solution. What is meant by 'target solution' is that the binary representation of the integer from the given vector never has adjacent bits which are the same.

It should be safe to make the assumption that binary representations are all the same length, so in general this length will be determined by the largest integer in the given vector. If that is case, it is straightforward to see that given a strings of bits of a fixed length, m, there are always exactly 2 target solutions. If we again assume that the input always contains two target numbers, then the particular case of a vector of length 4 will have 2 target solutions and 2 numbers that don't apply.

At the end of this report, we will give some thoughts on how this problem can be generalized. In the mean time, as far as the present task is concerned; we assume the following:

1. The input vector z has length 4,
2. the length of bits representing given integers is 4,
3. and here are exactly 2 target solutions

Given the above, my objective is to represent the integers on a quantum circuit and then output a linear superposition of target indices (i.e. two binary numbers from 00 to 11, represented as a superposition of computational basis states, that show the location of the target integers in the input vector.)

Strategy: amplitude amplification, a generalization of Grover's algorithm

With the formulation of the problem as above, my strategy is to use Amplitude amplification to obtain the desired superposition. This is a generalization of the popular Grover's algorithm. It increases the amplitude of target quantum vectors, and diminishes the amplitude of the other vectors.

I start by initializing a two-qubit quantum register into a superposition of all possible values. These qubits will be used to address the integers in the list, and the amplitude amplification algorithm will amplify the target qubits. The state vector of this register should be the output of my circuit.

The second stage is to load the integer values onto a quantum register. For this case of the task, this is a 16-qubit quantum register.

The next thing to do is to iterate the amplification algorithm. This consists of two steps: an oracle call, and a Grover diffusion operator. I designed the oracle to kick back a negative phase to the target address qubits. This is broken down into 3 steps:

1. Using the structure of a QRAM circuit, I load a superposition of the integer representation onto an output register, entangled to the address qubits.

2. I make an oracle call; a subroutine in the circuit that kicks back a phase to the address qubits.

3. I 'uncompute' the write procedure, to prepare the algorithm for the next instance.

The next step is to apply the Grover diffusion operator.

The procedure is iterated a few times, and I can examine the address register for results.

Simulation and results

I examine the results by looking at measurement statistics on the address register. Unfortunately, I did not obtain the desired results, and have so far been unsuccessful in troubleshooting where the algorithm or code fails. It appears the oracle is not working as expected, as the amplitudes remain evenly distributed. On the other hand, since I use a very large circuit to approach the task, I am not sure how much, if at all, simulation errors may have affected the output.

Extending the problem to general cases

I offer here a few thoughts for generalizing the solution. Firstly, the size of the given vector may change. In any case, I would always ensure the vector is of length 2^n, appending zeros if necessary. Also, the length of bits required to represent the integers may need to be modified.

In the general 2^n case, we will need n address qubits within which to search for target solutions. No matter the size of the system, there will always be 2 target solutions. The QRAM structure is not hard to generalize. The more challenging aspects will be the modifications to the oracle design and the Grover diffusion operator.

Overall, the amplitude amplification algorithm can be applied to all cases; the challenge will be to appropriately modify the circuit.

Conclusions

In conclusion, this task turned out to be quite challenging for a seemingly simple problem. Unfortunately, I was unable to execute the task as required.

I have included commentary in the Jupyter notebook that explains how each part of the algorithm is executed in the code.

Inspiration for the QRAM structure was taken from here. The algorithm applied is detailed here.