CIS 492/593 Lab 7 Grover's Search Algorithm Fall 2023

In this hands-on lab, you will learn and practice:

- the Grover's algorithm, including the Oracle construction and the Amplitude magnification method
- the Qiskit program for solving the 3-SAT problem via Grover's Search

Login to a workstation and open a terminal (either middle click the terminal icon on the left side bar or press CTRL-ALT-T). In the terminal window, type

```
cd QC
jupyter notebook
```

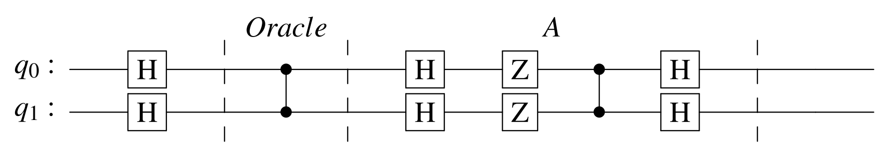
Inside the browser, click "New" and choose "Python 3 (ipykernel)". For each experiment, you need to put the experiment number as a comment (e.g. `# Experiment 1`) in the top of the code.

Experiment 1: Grover's Algorithm with a 2-bit oracle

- In the empty cell, type

  ```
  %load ~cis492s/pub/Grover2.py
  ```

  and click "Run" to load the program which will find the desired element location 11 among four binary strings: 00, 01, 10, and 11. In fact, this is a very simple SAT problem that determines the satisfiability of the expression $q_0 \wedge q_1$.
- Study the code. Note that the `Oracle()` function builds and returns the circuit which flips the sign of the probability amplitude of the location we are trying to find. The function `A()` performs the work of amplitude amplification by flipping a sequence of numbers about their mean. Below is the circuit generated from the program:



  Recall that

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad Controlled\text{-}Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

- We are interested in the operation matrices for the oracle and for the amplitude amplification circuits. Instead of calculating the matrices by hand, you can use the `Operator` class defined in the Qiskit `quantum_info` library. Add the following line of the code

  ```
  display(rdm(quantum_info.Operator(oc).data,n))
  ```

  to the `Oracle()` function before it returns. Also, Add the following line of the code

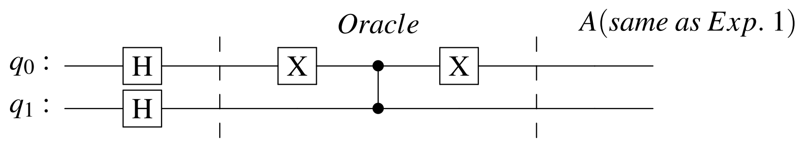
  ```
  display(rdm(quantum_info.Operator(ac).data,n))
  ```

  to the `A()` function before it returns. Because the Qiskit Operator uses a reverse bit sequence (i.e. $q_1 q_0$), we implement and call the function `rdm()` to reverse the bit sequence to be consistent with the order (i.e. $q_0 q_1$) used in the Bernhardt's textbook.
- Click the cell containing the code and then click "Run" to execute it. Record the matrix for the oracle. You may ignore the imaginary part (i.e. the term associated with the letter 'j'). Show which element location has the value `-1`. Note that `-1` will flip the probability amplitude of the element we are trying to find.

Record the matrix for the amplitude amplification. Also ignore the imaginary part. Is the matrix the same as the the matrix A in the textbook (page 179, line 18)?
Record and explain the result of the state vector (i.e. probability amplitude) (with the transpose operator).
What is the probability for each location?

Experiment 2: Grover's Algorithm with another 2-bit oracle

- Click the cell which contains the code done in Experiment 1. Click the "Edit" button and select "Copy Cells". Click "Edit" again and choose "Paste Cells Below".
- Instead of finding '11' as in the Experiment 1, you are asked to search the element location '01'. Note that this can be seen as a SAT problem that determines the satisfiability of the expression $\neg q_0 \wedge q_1$. To achieve this, you only need to add two X-gates, one is before the Controlled-Z and the other is after the the Controlled-Z, on the qubits that should be zero. In this experiment, it is the qubit q_0 which has to be enclosed by a pair of X-gates, just like the circuit below:



  Note that the first X-gate in the above diagram will swap the columns '00' and '10', and also swap the columns '01' and '11', in the diagonal matrix defined by the Controlled-Z, while the second X-gate will exchange the rows '00' and '10', and also exchange the rows '01' and '11'. These swappings will move the value `-1` to the desired element location.
  Modify the code in the `Oracle()` function based on the diagram above.
- Click "Run" to execute the code. Record the matrix for the oracle. Again, ignore the imaginary part. Indicate the location which contains the value `-1`.
  Record and explain the result of the state vector (i.e. probability amplitude) (with the transpose operator).
  What is the probability for each location?

Experiment 3: Grover's Algorithm with a 3-bit Oracle

- In the empty cell, type
  ```
  %load ~cis492s/pub/Grover3.py
  ```
  and click "Run" to load a program which will find the desired element location 111 among eight binary strings: 000, 001, 010, 011. 100, 101, 110, and 111. That is, this is a SAT problem that determines the satisfiability of the expression $q_0 \wedge q_1 \wedge q_2$.
  Qiskit does not support the ccz gate. As shown in the `Oracle()` function, we use a ccx gate and put a pair of H-gates before and after the X gate in ccx to emulate the ccz operation. Note that $HXH = Z$.
- Click "Run" to execute the code. Record the state vector result (with the transpose operator). Ignore the imaginary part. You should get two different probability amplitudes. Compare these two probability amplitudes with the two probability amplitudes mentioned in the textbook (page 180, line 4). Before you compare, the two probability amplitudes in the textbook should be divided by $\sqrt{8}$ due to $H^{\otimes 3}$.
- To see whether we can get a higher probability of the correct answer, we may flip the sign of the probability amplitude associated with the desired location and then perform the amplitude amplification again. That is, we need to execute the function `Oracle()` and `A()` again. This can be implemented by invoking the `repeat(iter)` method in Qiskit where `iter` is the number of repetition times. Modify

the code

```
# put the OA after the H-gates
qc.compose(OA, inplace=True)
```

to be

```
# put the repeated OA after the H-gates
qc.compose( OA.repeat(2), inplace=True)
```

- Click "Run" to execute the code.
  Record the result of the state vector (i.e. probability amplitude) (with the transpose operator).
  What is the probability for each location? Does the probability of the desired location become higher?
- Modify the code again to repeatedly execute the function `Oracle()` and `A()` three times. Record the result of the state vector (i.e. probability amplitude) (with the transpose operator). Does the probability of the desired location become higher? If not, that means we 'overcook' the numbers. Change the repetition times back to 2.

Experiment 4: Grover's Algorithm with another 3-bit Oracle

- Click the cell which contains the code done in Experiment 3. Click the "Edit" button and select "Copy Cells". Click "Edit" again and choose "Paste Cells Below".
- Using the method you learned in Experiment 2, modify the code in the `Oracle()` function to make the desired location be '001'. Namely, it can be seen as a SAT problem that determines the satisfiability of the expression $\neg q_0 \wedge \neg q_1 \wedge q_2$
- Click "Run" to execute the code. Record and explain the result of the state vector (i.e. probability amplitude) (with the transpose operator).

Experiment 5: Solving 3-SAT problems using Grover's Algorithm

Note that Grover's algorithm is not practical for most database searches. However, it is useful for solving the NP-complete problems such as 3-SAT, etc. and can give a quadratic speedup over classical algorithms.

Follow the steps below to learn how to use Qiskit to solve the 3-SAT problem with Grover's algorithm.

- Open the browser in a new window to browse:
  `https://github.com/Qiskit/qiskit-tutorials/blob/master/tutorials/algorithms/07_grover_examples.ipynb`
- Study the contents in the above web page.
- Copy the code from the web page to the cell in your jupyter and click "Run" to execute the code.
- Record the results.

Click "File" and choose "Save as". Type "lab7" in the entry box and click "Save" to save your work today into the file `lab7.ipynb`.

To turn in your file, use CTRL-ALT-T to open a terminal and type

```
ssh grail
```

and type your password to login to the server grail. Then, type

```
cd QC
turnin -c cis492s -p lab7 lab7.ipynb
```

to electronically submit your file `lab7.ipynb`.

Shutdown the jupyter notebook and logout the workstation.

Hand in your lab report before your leave.