Exercise3A

November 17, 2022

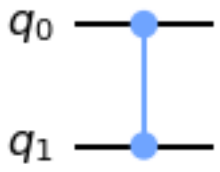
[1]:
```
import numpy as np

# Importing standard Qiskit libraries
from qiskit import QuantumCircuit, transpile, Aer, IBMQ
from qiskit.tools.jupyter import *
from qiskit.visualization import *
from ibm_quantum_widgets import *
from qiskit.providers.aer import QasmSimulator

# Loading your IBM Quantum account(s)
provider = IBMQ.load_account()
```

```
<frozen importlib._bootstrap>:219: RuntimeWarning:
scipy._lib.messagestream.MessageStream size changed, may indicate binary
incompatibility. Expected 56 from C header, got 64 from PyObject
```

[2]:
```
from qiskit import *
import matplotlib.pyplot as plt
import numpy as np
```

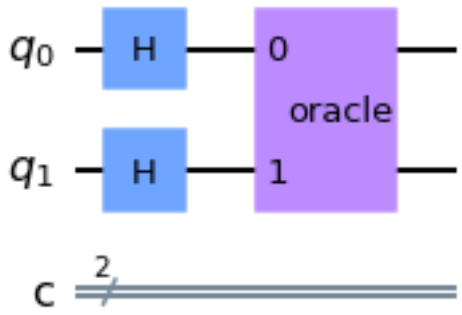
[3]:
```
## define the oracle circuit
oracle = QuantumCircuit(2,name="oracle")
oracle.cz(0,1)
oracle.to_gate()
oracle.draw(output="mpl")
```

[3]:



[4]:
```
backend = Aer.get_backend('statevector_simulator')
grover_circ = QuantumCircuit(2,2)
#hardamard gate on both of the qubits
grover_circ.h([0,1])

#then add on the oracle, so i can query each one of my states at the same time
grover_circ.append(oracle,[0,1])
grover_circ.draw(output='mpl')
```

[4]:



[5]:
```
job = execute(grover_circ, backend)
result = job.result()
sv = result.get_statevector()
np.around(sv,2)
```

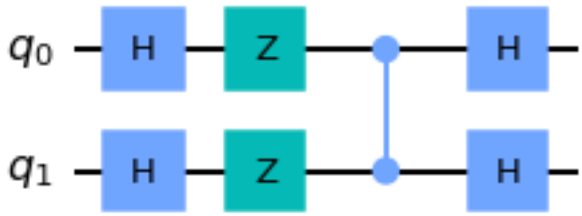
[5]:
```
array([ 0.5+0.j,  0.5+0.j,  0.5+0.j, -0.5+0.j])
```

[6]:
```
reflection = QuantumCircuit(2, name='reflection')
reflection.h([0,1])
reflection.z([0,1])
reflection.cz(0,1)
# this is a circuit that will apply a negative phase only to 00 state.
# and then we transform it back with a hadamard with both qubits.
```

[6]:
```
<qiskit.circuit.instructionset.InstructionSet at 0x7fa279078b20>
```

[7]:
```
reflection.h([0,1])
reflection.to_gate()
reflection.draw(output='mpl')
```

[7]:



[8]:
```
backend = Aer.get_backend('qasm_simulator')
grover_circ = QuantumCircuit(2,2) # two qubits two classical registers
```

[9]:
```
#prepare superposition state, hadamards on both of the qubits
grover_circ.h([0,1])
grover_circ.append(oracle,[0,1])
```

[9]: <qiskit.circuit.instructionset.InstructionSet at 0x7fa278fc0d30>

[10]:
```
grover_circ.append(reflection,[0,1])
```

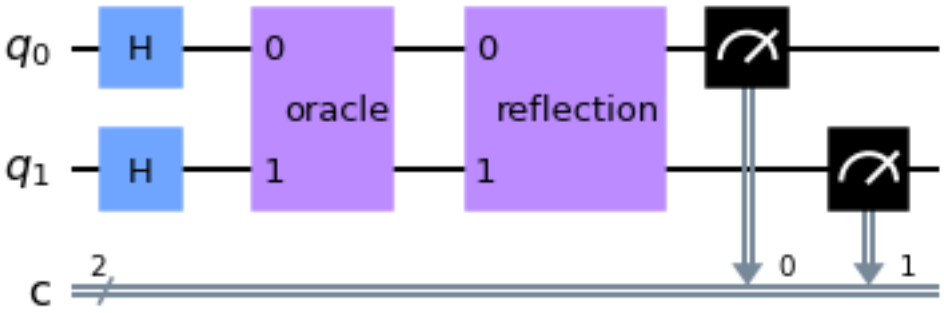
[10]: <qiskit.circuit.instructionset.InstructionSet at 0x7fa2765860d0>

[11]:
```
grover_circ.measure([0,1],[0,1])
```

[11]: <qiskit.circuit.instructionset.InstructionSet at 0x7fa278feafa0>

[12]:
```
grover_circ.draw(output='mpl')
```

[12]:



[13]:
```
job = execute(grover_circ, backend, shots=1)
result = job.result()
result.get_counts()
```

[13]:
```
{'11': 1}
```

[ ]:
```

```