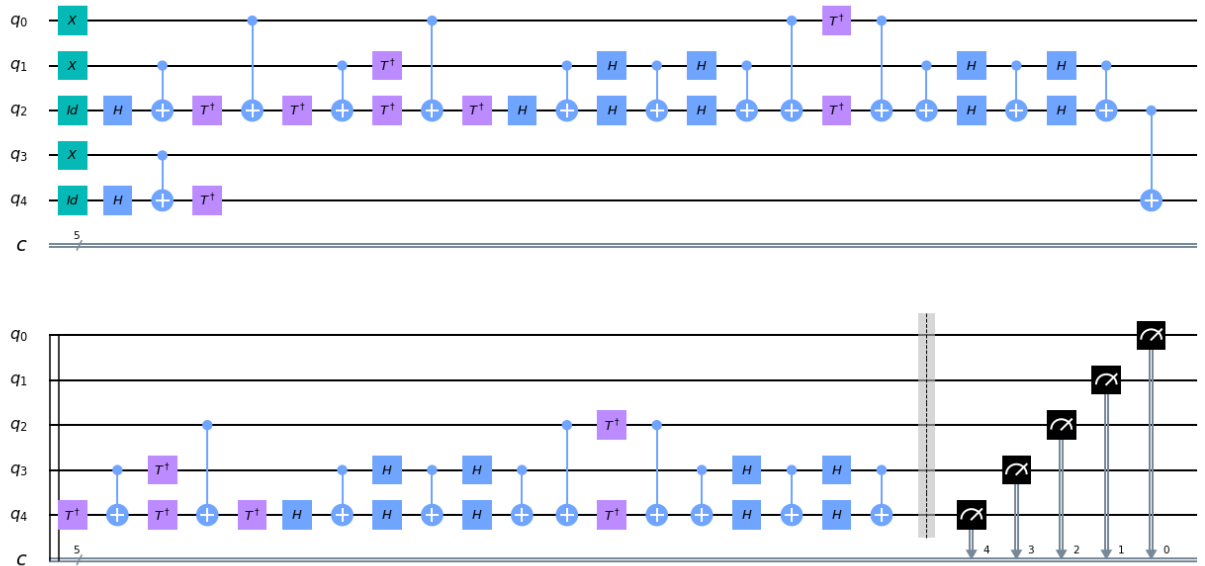


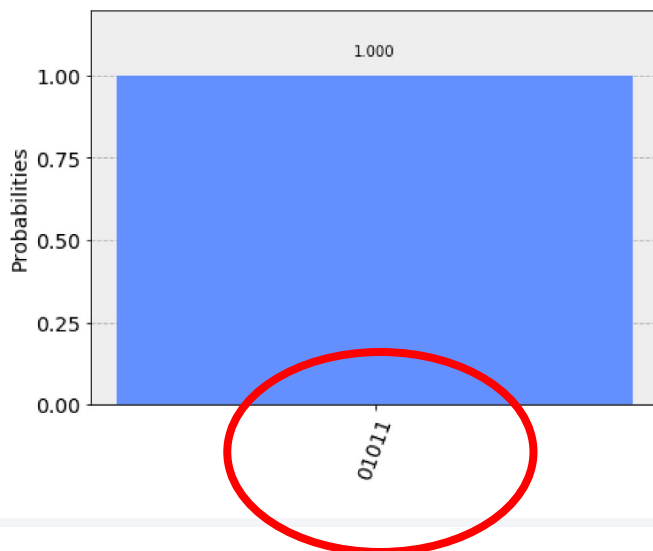
```
In [13]: qc = QuantumCircuit(5,5)
          qc.x(0)
          qc.x(1)
          qc.x(3)
          qc.iden(2)
          qc.h(2)
          qc.cx(1,2)
          qc.tdg(2)
          qc.cx(0,2)
          qc.tdg(2)
          qc.cx(1,2)
          qc.tdg(2)
          qc.cx(0,2)
          qc.tdg(1)
          qc.tdg(2)
          qc.h(2)
          qc.cx(1,2)
          qc.h(1)
          qc.h(2)
          qc.cx(1,2)
          qc.h(1)
          qc.h(2)
          qc.cx(1,2)
          qc.cx(0,2)
          qc.tdg(2)
          qc.tdg(0)
          qc.cx(0,2)
          qc.cx(1,2)
          qc.h(1)
          qc.h(2)
          qc.cx(1,2)
          qc.h(1)
          qc.h(2)
          qc.cx(1,2)
          qc.iden(4)
          qc.h(4)
          qc.cx(3,4)
          qc.tdg(4)
          qc.cx(2,4)
          qc.tdg(4)
          qc.cx(3,4)
          qc.tdg(4)
          qc.cx(2,4)
          qc.tdg(3)
          qc.tdg(4)
          qc.h(4)
          qc.cx(3,4)
          qc.h(3)
          qc.h(4)
          qc.cx(3,4)
          qc.h(3)
          qc.h(4)
          qc.cx(3,4)
          qc.h(3)
          qc.h(4)
          qc.cx(3,4)
          qc.cx(2,4)
          qc.tdg(4)
          qc.tdg(2)
          qc.cx(2,4)
          qc.cx(3,4)
          qc.h(3)
          qc.h(4)
          qc.cx(3,4)
          qc.h(3)
          qc.h(4)
          qc.cx(3,4)
          qc.barrier(range(5))
          qc.measure(4,4)
          qc.measure(3,3)
          qc.measure(2,2)
          qc.measure(1,1)
          qc.measure(0,0)
          qc.draw()
```



```
In [14]: # Use Aer's qasm_simulator
backend_sim = Aer.get_backend('qasm_simulator')

# Execute the circuit on the qasm simulator.
# We've set the number of repeats of the circuit
# to be 1024, which is the default.
job_sim = execute(qc, backend_sim, shots=1024)

# Grab the results from the job.
result_sim = job_sim.result()
counts = result_sim.get_counts(qc)
#print(counts)
plot_histogram(counts)
```



```
In [ ]:
```