

Quantum Pratical Assignment



Grupo 23

Elementos:

- a87982 Bruna Dias Carvalho
- a82124 Maria Luísa Faria Silva



Conteúdo

1. Algoritmo de Procura
2. Simulação de Ruído
3. Execução em Computador Quântico
4. IGNIS

```
In [1]: # importing Qiskit
from qiskit import Aer, IBMQ
from qiskit import QuantumCircuit, ClassicalRegister, QuantumRegister
from qiskit import execute, transpile

from qiskit.tools.visualization import plot_histogram, plot_state_city, plot_state_hinton

import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [2]: s = 23 % 8
print(s)
```

7

```
In [3]: wb = bin(s)[2:]
print(wb)
```

111

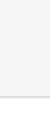
```
In [4]: x = len(wb)
print('number of qubits: ', x)
```

number of qubits: 3

Primeiro começamos por criar os qubits necessário para representar o nosso número, ou seja, x qubits

```
In [5]: qr_x = QuantumRegister(x, 'x')
backend = Aer.get_backend("qasm_simulator")
cr=ClassicalRegister(x,'cr')
qc_Grover=QuantumCircuit(qr_x,cr)
```

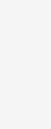
Algoritmo



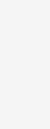
Este algoritmo de procura(*Grover's Algorithm*) numa lista não ordenada encontra-se dividido em 3 fases, inicialização, *oracle* e amplificação.

Inicialização do sistema com a mesma amplitude em todos os *input states* possíveis

$$\sum_{x_i} |x_i\rangle$$

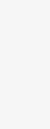


```
In [6]: # Init
qc_Grover.h(0)
qc_Grover.h(1)
qc_Grover.h(2)
```



```
Out[6]: <qiskit.circuit.instructionset.InstructionSet at 0x17cfcd6940>
```

Aplica \sqrt{N} vezes a seguinte operação:



Quantum Oracle operator U_w . Este operador é responsável por identificar a solução para o problema e indicar o objetivo.

$$-\alpha_m |x_m\rangle + \beta \sum_{x_i \neq x_m} |x_i\rangle$$

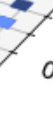


Com esta implementação, a fase do estado ($f(x_m) = 1$) roda π radians, enquanto os outros estados não sofrem alterações.

```
In [7]: import math as m
times = round(m.sqrt(2**x))
print(times)
```

3

Posteriormente, aplicamos um *diffuser* com intuito de uma amplificação, ou seja, para que a probabilidade do x_m aumente e as dos outros estados diminuam.



```
In [8]: def select_w(circuit, qr_x):
# Estamos a procura do estado |111>
# Logo não é preciso aplicar x
pass

def phase_oracle(circuit, qr_x):
select_w(circuit, qr_x)
circuit.h(0)
circuit.ccx(2, 1, 0)
circuit.h(0)
select_w(circuit, qr_x)

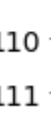
def diffuser(circuit, qr_x):
circuit.h(qr_x)
circuit.x(qr_x)

circuit.h(qr_x)
circuit.ccx(2, 1, 0)
circuit.h(0)

circuit.x(qr_x)
circuit.h(qr_x)

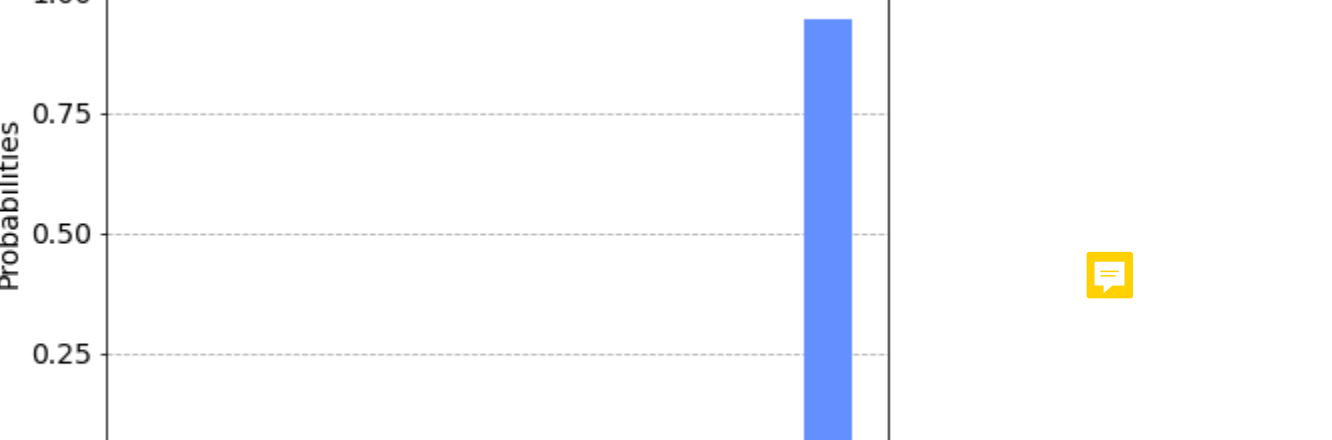
#oracle and diffuser
for t in range(2):
# phase oracle 2
phase_oracle(qc_Grover, qr_x)
# diffuser
diffuser(qc_Grover, qr_x)

qc_Grover.measure(qr_x, cr)
```



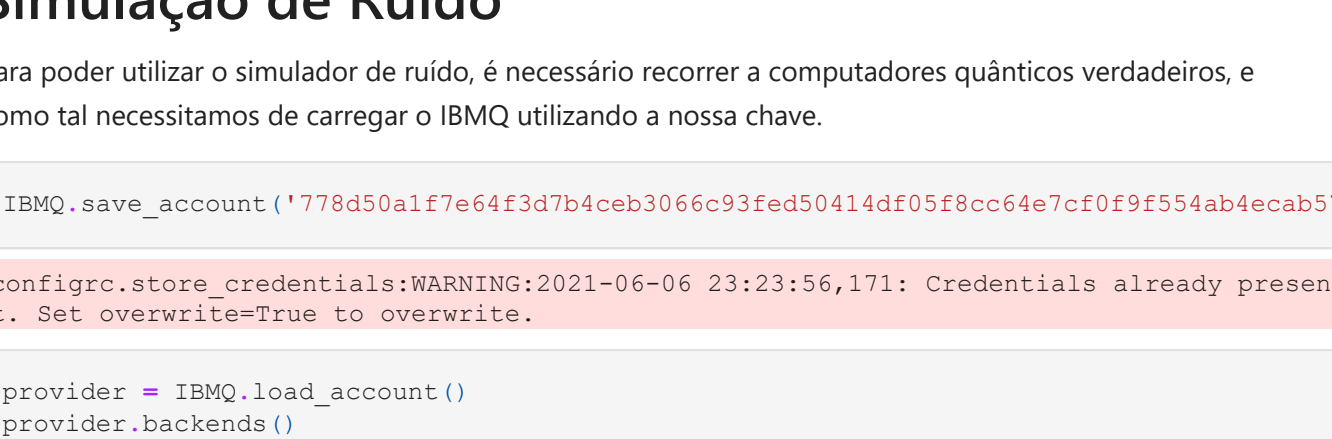
```
qc_Grover.draw(output='mpl')
```

```
Out[8]:
```



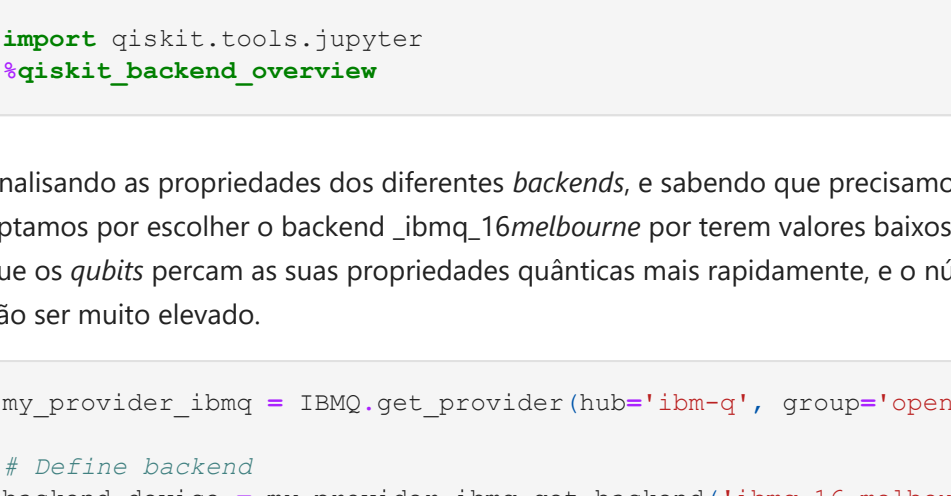
```
In [9]: backend_state = Aer.get_backend('statevector_simulator') # the device to run on
result = execute(qc_Grover, backend_state).result()
ps11 = result.get_statevector(qc_Grover)
plot_state_city(ps11)
```

```
Out[9]:
```



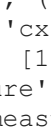
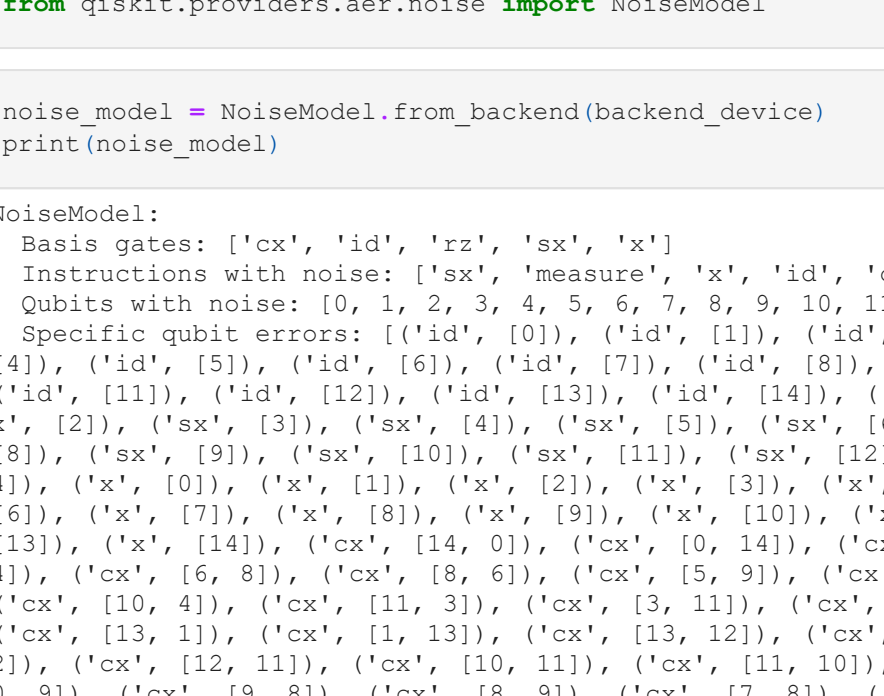
```
In [10]: plot_state_hinton(ps11)
```

```
Out[10]:
```



```
In [11]: backend_state = Aer.get_backend('qasm_simulator') # the device to run on
shots=1024
result = execute(qc_Grover, backend, shots=shots).result()
counts_sim = result.get_counts(qc_Grover)
plot_histogram(counts_sim)
```

```
Out[11]:
```



```
In [12]: qc_Grover.depth()
```

```
Out[12]: 22
```

Simulação de Ruído

Para poder utilizar o simulador de ruído, é necessário recorrer a computadores quânticos verdadeiros, e como tal necessitamos de carregá o IBMQ utilizando a nossa chave.

```
In [13]: IBMQ.save_account('778d50alf7e64f3d7b4ceb3066c93fed50414df05f8cc64e7cf0f9f554ab4ecab5')
```

```
configure_store_credentials:WARNING:2021-06-06 23:23:56,171: Credentials already present. Set overwrite=True to overwrite.
```

```
In [14]: provider = IBMQ.load_account()
provider.backends()
```

```
Out[14]: [<IBMQSimulator('ibmq_qasm_simulator') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQBackend('ibmqx2') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQBackend('ibmq_16_melbourne') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQBackend('ibmq_armonk') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQBackend('ibmq_athens') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQBackend('ibmq_santiago') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQBackend('ibmq_lima') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQBackend('ibmq_belem') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQBackend('ibmq_quito') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQSimulator('simulator_statevector') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQSimulator('simulator_mps') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQSimulator('simulator_extended_stabilizer') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQSimulator('simulator_stabilizer') from IBMQ(hub='ibm-q', group='open', project='main')>,
<IBMQBackend('ibmq_manila') from IBMQ(hub='ibm-q', group='open', project='main')>]
```

```
In [15]: import qiskit.tools.jupyter
%qiskit_backend_overview
```

Analisando as propriedades dos diferentes *backends*, e sabendo que precisamos no mínimo de 3 qubits optamos por escolher o backend *_ibmq_16melbourne* por terem valores baixos de *AVG T1/T2*, de modo a que os *qubits* percam as suas propriedades quânticas mais rapidamente, e o número de *jobs* pendentes não ser muito elevado.

```
In [16]: my_provider_ibmq = IBMQ.get_provider(hub='ibm-q', group='open', project='main')
```

```
# Define backend
backend_device = my_provider_ibmq.get_backend('ibmq_16_melbourne')
```

```
In [17]: %qiskit_job_watcher
```

```
Out[17]:
```

```
coupling_map = backend_device.configuration().coupling_map
```

```
In [18]:
```

```
coupling_map = backend_device.configuration().coupling_map
```

```
In [19]: from qiskit.providers.aer.noise import NoiseModel
```

```
In [20]: noise_model = NoiseModel.from_backend(backend_device)
print(noise_model)
```

```
NoiseModel:
  Basis gates: ['cx', 'id', 'rz', 'sx', 'x']
  Instructions with noise: ['sx', 'measure', 'x', 'id', 'cx']
  Qubits with noise: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
  Specific qubit errors: (('id', [0]), ('id', [1]), ('id', [2]), ('id', [3]), ('id', [4]), ('id', [5]), ('id', [6]), ('id', [7]), ('id', [8]), ('id', [9]), ('id', [10]), ('id', [11]), ('id', [12]), ('id', [13]), ('id', [14]), ('sx', [0]), ('sx', [1]), ('sx', [2]), ('sx', [3]), ('sx', [4]), ('sx', [5]), ('sx', [6]), ('sx', [7]), ('sx', [8]), ('sx', [9]), ('sx', [10]), ('sx', [11]), ('sx', [12]), ('sx', [13]), ('sx', [14]), ('x', [0]), ('x', [1]), ('x', [2]), ('x', [3]), ('x', [4]), ('x', [5]), ('x', [6]), ('x', [7]), ('x', [8]), ('x', [9]), ('x', [10]), ('x', [11]), ('x', [12]), ('x', [13]), ('x', [14]), ('cx', [0, 1]), ('cx', [1, 2]), ('cx', [2, 3]), ('cx', [3, 4]), ('cx', [4, 5]), ('cx', [5, 6]), ('cx', [6, 7]), ('cx', [7, 8]), ('cx', [8, 9]), ('cx', [9, 10]), ('cx', [10, 11]), ('cx', [11, 12]), ('cx', [12, 13]), ('cx', [13, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx', [14, 7]), ('cx', [14, 8]), ('cx', [14, 9]), ('cx', [14, 10]), ('cx', [14, 11]), ('cx', [14, 12]), ('cx', [14, 13]), ('cx', [14, 14]), ('cx', [14, 0]), ('cx', [14, 1]), ('cx', [14, 2]), ('cx', [14, 3]), ('cx', [14, 4]), ('cx', [14, 5]), ('cx', [14, 6]), ('cx
```