

量子计算理论基础与软件系统 实验报告

Lab 2 Quantum Algorithm

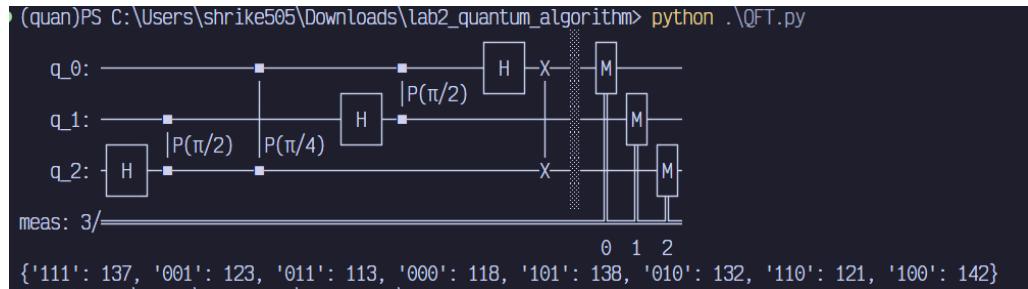
12.11.2025

redacted

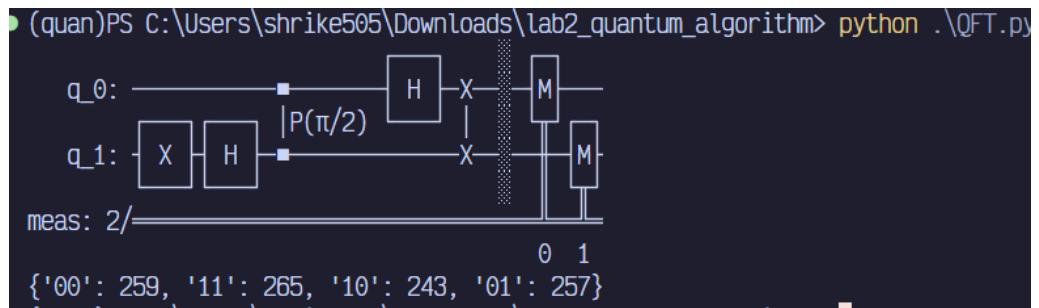
redacted

1. QFT 算法

查看所给三量子比特的 QFT 算法的频率分布结果：



接下来将 `n_qubits` 改为 2，并将初始态设置为 $|10\rangle$ （通过 `qc.x(1)`），得到运行结果如下：



理论上，对于量子态为 $|10\rangle$ 的双量子比特，输入 QFT 后有

$$\text{QFT}|10\rangle = \text{QFT}|2\rangle = \frac{1}{2} \sum_{k=0}^3 e^{2\pi i \cdot 2 \cdot \frac{k}{4}} |k\rangle = \frac{1}{2}(|0\rangle - |1\rangle + |2\rangle - |3\rangle) = \frac{1}{2}(|00\rangle - |01\rangle + |10\rangle - |11\rangle)$$

因此测量结果 00、01、10、11 的概率均为 $\frac{1}{4}$ ，与实验结果相符。

对于 `qft` 函数，记量子比特数为 `n`

```
1 for i in range(qc.num_qubits - 1, -1, -1): # 外层循环 n 次
2     qc.h(i)                                     # 1 层 H 门
3     for j in range(i - 1, -1, -1):             # 内层循环 i 次
4         qc.cp(pi / 2 ** (i - j), j, i)       # 受控相位门
5     for i in range(qc.num_qubits // 2):          # 交换操作
6         qc.swap(i, qc.num_qubits - i - 1)
```

对于第 `i` 个量子比特，需要与前面 `i-1` 个量子比特分别施加受控相位门。这些门如果控制位和目标位都不冲突，可以与某一个 H 门在同一层执行；而后面的受控相位门依赖于前面的 H 门，即 H 门不能并行执行；最后的交换操作显然只需要 $n//2$ 层。

由此可见，总的量子门层数为 $O(n)$ 。

2. QPE 算法

2.1. qpe_circuit

根据 QPE 算法的原理，构造 QPE 电路：

```

1  def qpe_circuit(theta, n_counting_qubits=3, eigenstate=None):    python
2      """
3          构建 QPE 电路
4          theta: 相位角度
5          n_counting_qubits: 计数寄存器量子比特数
6          eigenstate: 特征寄存器初态 [alpha, beta], None 表示使用本征态 |1>
7          """
8          # 创建寄存器
9          qr_counting = QuantumRegister(n_counting_qubits, 'counting')
10         qr_eigen = QuantumRegister(1, 'eigen')
11         cr = ClassicalRegister(n_counting_qubits, 'c')
12         qc = QuantumCircuit(qr_counting, qr_eigen, cr)
13
14         # 初始化特征寄存器
15         if eigenstate is None:
16             # 使用本征态 |1>
17             qc.x(qr_eigen[0])
18         else:
19             # 使用任意态 [alpha, beta]
20             alpha, beta = eigenstate
21             qc.initialize([alpha, beta], qr_eigen[0])
22
23         # 第一步：计数寄存器叠加
24         for i in range(n_counting_qubits):
25             qc.h(qr_counting[i])
26
27         # 第二步：受控酉操作
28         for i in range(n_counting_qubits):
29             # 应用  $2^i$  次受控相位门
30             control_qubit = qr_counting[i]
31             qc.cp(2**i * theta, control_qubit, qr_eigen[0])
32
33         # 第三步：逆 QFT
34         qc = qc.compose(QFT(n_counting_qubits, inverse=True), qr_counting)

```

```

35
36     # 测量计数寄存器
37     qc.measure(qr_counting, cr)
38
39     return qc

```

传入 $\theta = \pi/4$, $n_{\text{counting_qubits}} = 3$, 得到 QPE 电路输出频率分布如下:

- (quan)PS C:\Users\shrike505\Downloads\lab2_quantum_algorithm> python .\QPE.py
 C:\Users\shrike505\Downloads\lab2_quantum_algorithm\QPE.py:38: DeprecationWarning: The class ``qiskit.circuit.library.basis_change.qft.QFT`` has been deprecated since qiskit 3.0. (Use qiskit.circuit.library.QFTGate or qiskit.synthesis.qft.synth_qft_full instead, for access to all previous arguments.)
 qc = qc.compose(QFT(n_counting_qubits, inverse=True), qr_counting)

QPE 电路:

QPE 测量结果:
 $\{'001': 1024\}$

001 确实对应了相位 $1/8$ 。

2.2. 更改角度

将 θ 改为 $2\pi/3$, 得到输出频率分布如下:

- (quan)PS C:\Users\shrike505\Downloads\lab2_quantum_algorithm> python .\QPE.py
 C:\Users\shrike505\Downloads\lab2_quantum_algorithm\QPE.py:38: DeprecationWarning: The class ``qiskit.circuit.library.basis_change.qft.QFT`` has been deprecated since qiskit 3.0. (Use qiskit.circuit.library.QFTGate or qiskit.synthesis.qft.synth_qft_full instead, for access to all previous arguments.)
 qc = qc.compose(QFT(n_counting_qubits, inverse=True), qr_counting)

QPE 电路:

QPE 测量结果:
 $\{'010': 184, '011': 703, '100': 53, '001': 39, '000': 11, '110': 9, '101': 19, '111': 6\}$

理论上, 真实相位为 $1/3$, 与之最接近的状态 011 (相位 0.375) 出现频率最高: $703/1024 \approx 0.686$, 状态 010 (相位 0.25) 次高: $184/1024 \approx 0.180$, 其他状态的出现概率很低, 符合误差分布, 这与理论一致。

2.3. 复杂度计算

初始化 H 门层数时，所有 H 门可以并行；对于受控酉操作层数，第 i 个量子比特需要 2^i 次受控相位门，这些门可以部分并行，但受限于控制-目标对的冲突；根据之前的分析，逆 QFT 需要 $O(n)$ 层

因此总的复杂度为 $O(n)$ 。

2.4. 更改初态

传入非本征态初始向量 $[3/5, 4/5]$ ，得到输出频率分布如下：

```
(quan)PS C:\Users\shrike505\Downloads\lab2_quantum_algorithm> python .\QPE.py
C:\Users\shrike505\Downloads\lab2_quantum_algorithm\QPE.py:38: DeprecationWarning: The class ``qiskit.circui
3.0. ('Use qiskit.circuit.library.QFTGate or qiskit.synthesis.qft.synth_qft_full instead, for access to all
qc = qc.compose(QFT(n_counting_qubits, inverse=True), qr_counting)
QPE 电路:
  counting_0: ┌───┐
              H
  counting_1: ┌───┐
              H
  counting_2: ┌───┐
              H
  eigen:   ┌─────────┐
           Initialize(0.6,0.8)
  c: 3/    └─────────┘
                  P(2π/3)  P(4π/3)  P(8π/3)
               ┌─────────┐ ┌─────────┐ ┌─────────┐
               0          1          2
               ┌───┐ ┌───┐ ┌───┐
               M  M  M
               └───┘ └───┘ └───┘
测量结果:
状态 000: 379 次, 估计相位 = 0.000
状态 001: 22 次, 估计相位 = 0.125
状态 010: 114 次, 估计相位 = 0.250
状态 011: 448 次, 估计相位 = 0.375
状态 100: 33 次, 估计相位 = 0.500
状态 101: 13 次, 估计相位 = 0.625
状态 110: 6 次, 估计相位 = 0.750
状态 111: 9 次, 估计相位 = 0.875
理论相位: 0.333
```

理论上，初态中 $|0\rangle$ 分量的概率为 $\frac{9}{25}$ ， $|1\rangle$ 分量的概率为 $\frac{16}{25}$ ；QPE 作用于非本征态时， $|0\rangle$ 分量对应的相位为 0， $|1\rangle$ 分量对应的相位为 $1/3$ 。因此测量结果中，状态 000（相位 0）出现概率约为 $\frac{9}{25} = 0.36$ ，状态 011 和 010 应该有较高的概率，这和实验结果一致（000 出现频率为 0.37，011 和 010 出现频率分别约为 0.446 和 0.114）。

3. Shor 算法

3.1. 补全代码

```
1 def mod_circuit(a, N, n_v):
2     matrix = np.zeros((2 ** n_v, 2 ** n_v), dtype=int)
3     for y in range(2 ** n_v):
4         if y < N:
5             x = (a * y) % N
```

python

```

6         matrix[x, y] = 1
7     else:
8         matrix[y, y] = 1
9     return UnitaryGate(matrix)

```

● (quan)PS C:\Users\shrike505\Downloads\lab2_quantum_algorithm> python .\Shor.py
C:\Users\shrike505\Downloads\lab2_quantum_algorithm\Shor.py:18: DeprecationWarning: The class ``qiskit.circuit.library.QFTGate or qiskit.synthesis.qft.synth_qft_full instead, qc.append(QFT(n_p, inverse=True), range(n_p))

counts: {'110': 240, '000': 266, '100': 258, '010': 260}
r: 4
15 = 3 * 5

成功分解 $15 = 3 \times 5$ 。

3.2. 分解 21

● (quan)PS C:\Users\shrike505\Downloads\lab2_quantum_algorithm> python .\Shor.py
C:\Users\shrike505\Downloads\lab2_quantum_algorithm\Shor.py:18: DeprecationWarning: The class ``qiskit.circuit.library.QFTGate or qiskit.synthesis.qft.synth_qft_full instead, for access to all previous arguments.'' qc.append(QFT(n_p, inverse=True), range(n_p))

counts: {'10000': 513, '00000': 511}
r: 2
21 = 7 * 3

命 $a = 8$, $n_p = n_v = 5$, 成功分解 $21 = 3 \times 7$ 。