量子计算理论基础与软件系统 实验报告

Lab 2 Quantum Algorithm

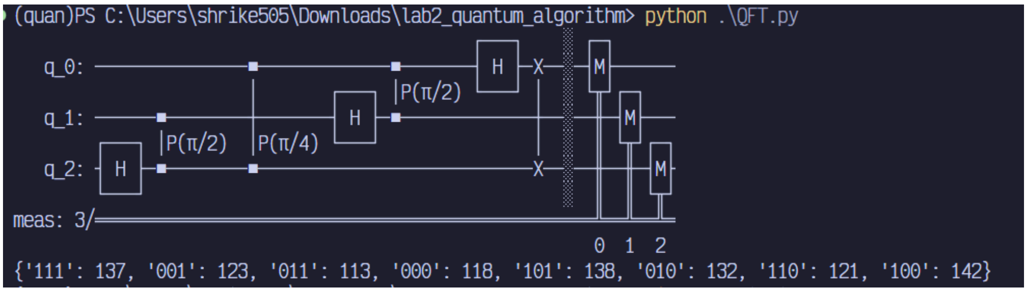
12.11.2025

redacted

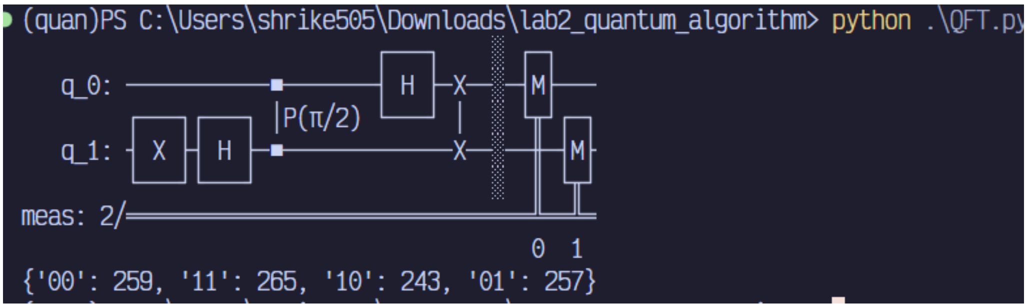
redacted

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⟩`（通过 `qc.x(1)`），得到运行结果如下：



理论上，对于量子态为 `|10⟩` 的双量子比特，输入 QFT 后有

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

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`

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

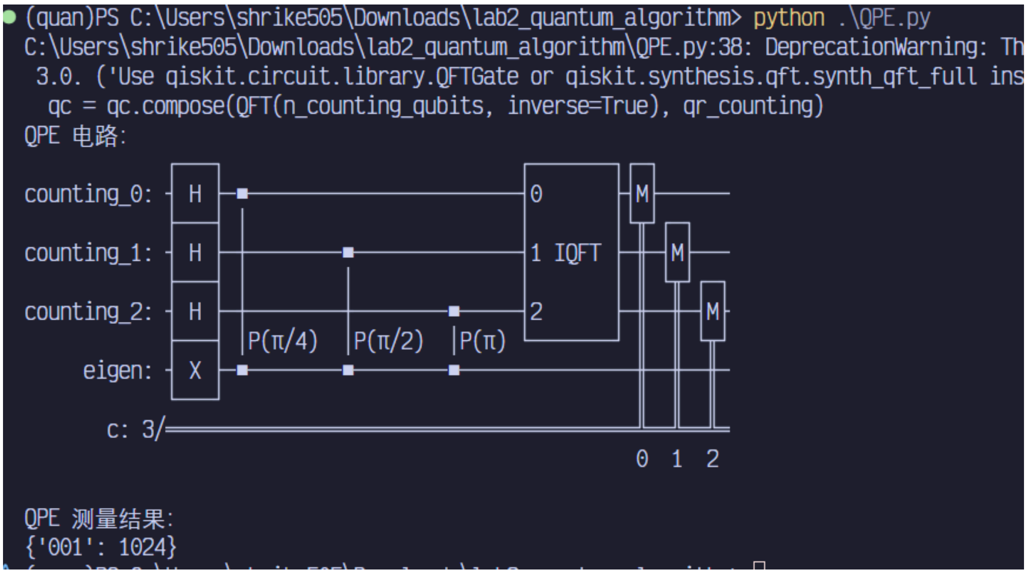
2.1. qpe_circuit

根据 QPE 算法的原理，构造 QPE 电路：

```python
def qpe_circuit(theta, n_counting_qubits=3, eigenstate=None):
    """
    构建 QPE 电路
    theta: 相位角度
    n_counting_qubits: 计数寄存器量子比特数
    eigenstate: 特征寄存器初态 [alpha, beta], None 表示使用本征态 |1>
    """
    # 创建寄存器
    qr_counting = QuantumRegister(n_counting_qubits, 'counting')
    qr_eigen = QuantumRegister(1, 'eigen')
    cr = ClassicalRegister(n_counting_qubits, 'c')
    qc = QuantumCircuit(qr_counting, qr_eigen, cr)

    # 初始化特征寄存器
    if eigenstate is None:
        # 使用本征态 |1>
        qc.x(qr_eigen[0])
    else:
        # 使用任意态 [alpha, beta]
        alpha, beta = eigenstate
        qc.initialize([alpha, beta], qr_eigen[0])

    # 第一步：计数寄存器叠加
    for i in range(n_counting_qubits):
        qc.h(qr_counting[i])

    # 第二步：受控酉操作
    for i in range(n_counting_qubits):
        # 应用 2^i 次受控相位门
        control_qubit = qr_counting[i]
        qc.cp(2**i * theta, control_qubit, qr_eigen[0])

    # 第三步：逆 QFT
    qc = qc.compose(QFT(n_counting_qubits, inverse=True), qr_counting)
```

```

    # 测量计数寄存器
    qc.measure(qr_counting, cr)

    return qc
```

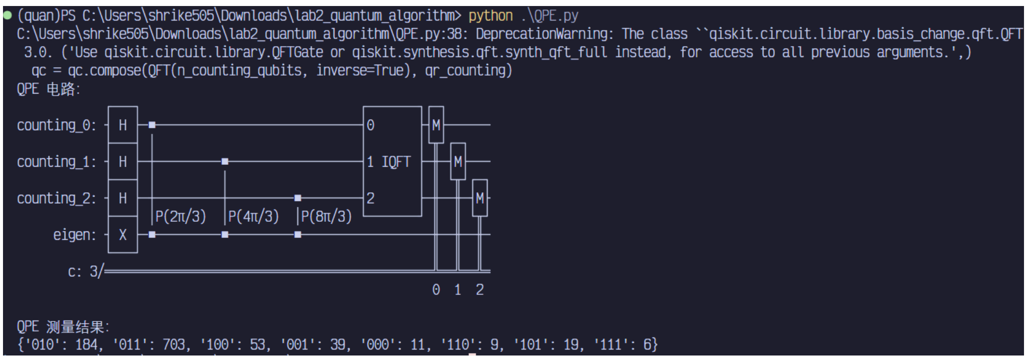
传入 theta = π/4， n_counting_qubits = 3， 得到 QPE 电路输出频率分布如下：



001 确实对应了相位 1/8。

2.2. 更改角度

将 theta 改为 2π/3， 得到输出频率分布如下：



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 ≈ 0.686，状态 010（相位 0.25）次高：184/1024 ≈ 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。

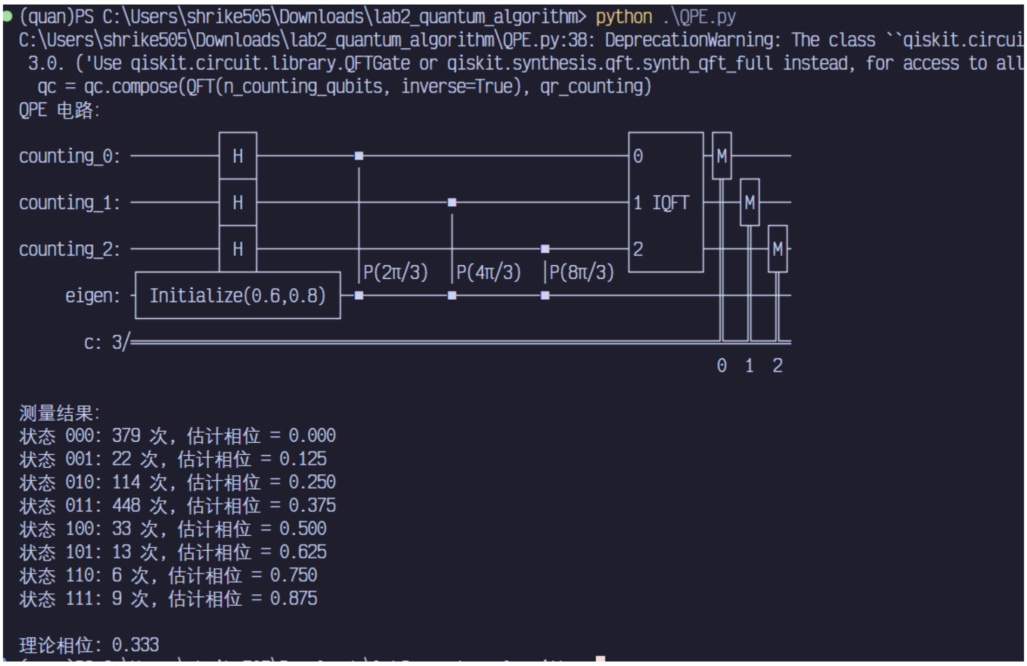
2.3. 复杂度计算

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：



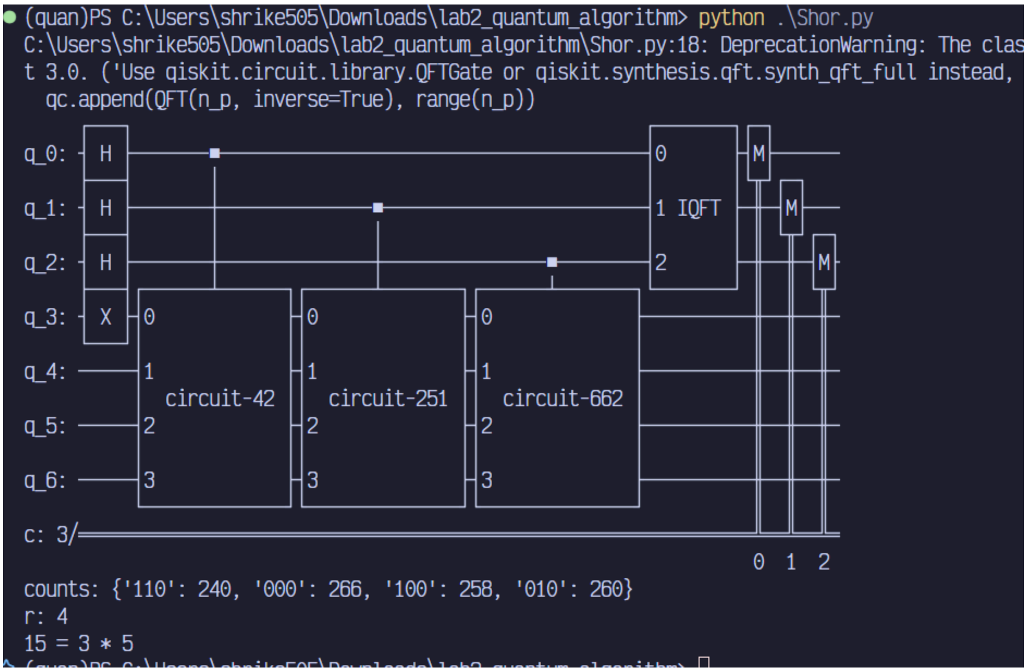
理论上，初态中 $|0\rangle$ 分量的概率为 $\frac{9}{25}$，$|1\rangle$ 分量的概率为 $\frac{16}{25}$；QPE 作用于非本征态时，$|0\rangle$ 分量对应的相位为 0，$|1\rangle$ 分量对应的相位为 `1/3`。因此测量结果中，状态 `000`（相位 0）出现概率约为 $\frac{9}{25} = 0.36$，状态 `011` 和 `010` 应该有较高的概率，这和实验结果一致（`000` 出现频率为 0.37，`011` 和 `010` 出现频率分别约为 0.446 和 0.114）。

3. Shor 算法

3.1. 补全代码

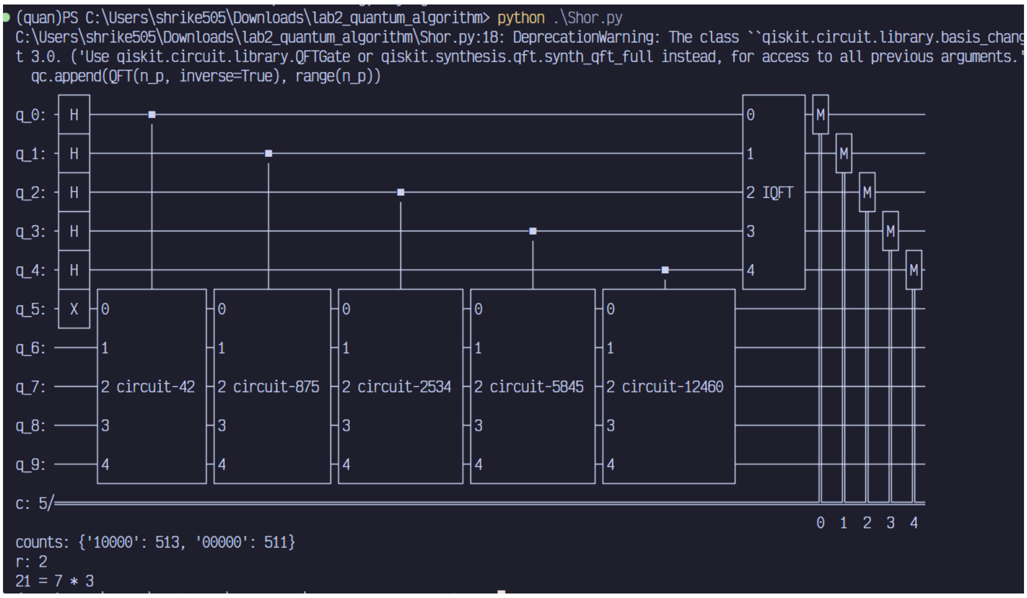
```python
def mod_circuit(a, N, n_v):
    matrix = np.zeros((2 ** n_v, 2 ** n_v), dtype=int)
    for y in range(2 ** n_v):
        if y < N:
            x = (a * y) % N
```

```
            matrix[x, y] = 1
        else:
            matrix[y, y] = 1
    return UnitaryGate(matrix)
```



成功分解 $15 = 3 \times 5$。

3.2. 分解 21



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