소인수 분해 알고리즘 완전정복

09.

쇼어 알고리즘의 양자회로 구현 (번외편)

9. 쇼어 알고리즘의 양자회로 구현

주니온TV@Youtube
자세히 보면 유익한 코딩 채널

- **쇼어의 소인수 분해 알고리즘**
  1. 1보다 크고 N보다 작은 정수 a를 임의로 선택
  2. 만일, $\gcd(N, a) \neq 1$이면, p를 찾았다!
     - 따라서, $p = \gcd(N, a), q = N/\gcd(N, a)$
  3. 함수 $f(x) = a^x \pmod N$의 **주기(period) r을 찾는다.**
     - 여기서 찾은 주기 r이 짝수가 아니면, 1번 단계부터 다시 시작한다.
  4. 주기 r로부터 두 개의 최대공약수 gcd_1, gcd_2 를 찾는다.
     - $gcd_1 = \gcd(N, a^{r/2} + 1), gcd_2 = \gcd(N, a^{r/2} - 1)$
  5. gcd_1, gcd_2이 1과 N이라면, 1번 단계부터 다시 시작한다.
     - 아니면, 마침내 소인수들을 찾았으므로, $p = gcd_1, q = gcd_2$ 리턴

9. 쇼어 알고리즘의 양자회로 구현

- 쇼어의 소인수 분해 알고리즘

```
def factorize4(N):
    while(True):
        a = random.randint(2, N - 1)
        gcd = math.gcd(N, a)
        if (gcd != 1):
            return gcd, N // gcd
        r = findPeriodByQuantumCircuit(N, a)
        if (r % 2 != 0):
            continue
        gcd1 = math.gcd(N, a**(r//2) + 1)
        gcd2 = math.gcd(N, a**(r//2) - 1)
        if (gcd1 == 1 or gcd2 == 1):
            continue
        return gcd1, gcd2
```

9. 쇼어 알고리즘의 양자회로 구현

- **함수 $f(x) = a^x \pmod N$의 주기 찾기**
  - **양자 위상 추정**: Quantum Phase Estimation
  - **연분수 확장**: Continued Fraction

```
def findPeriodByQuantumCircuit(N, a):
    phase, qc = qpe_amod15(a)
    frac = Fraction(phase).limit_denominator(15) # 분모가 15
    return frac.denominator, qc
```

9. 쇼어 알고리즘의 양자회로 구현

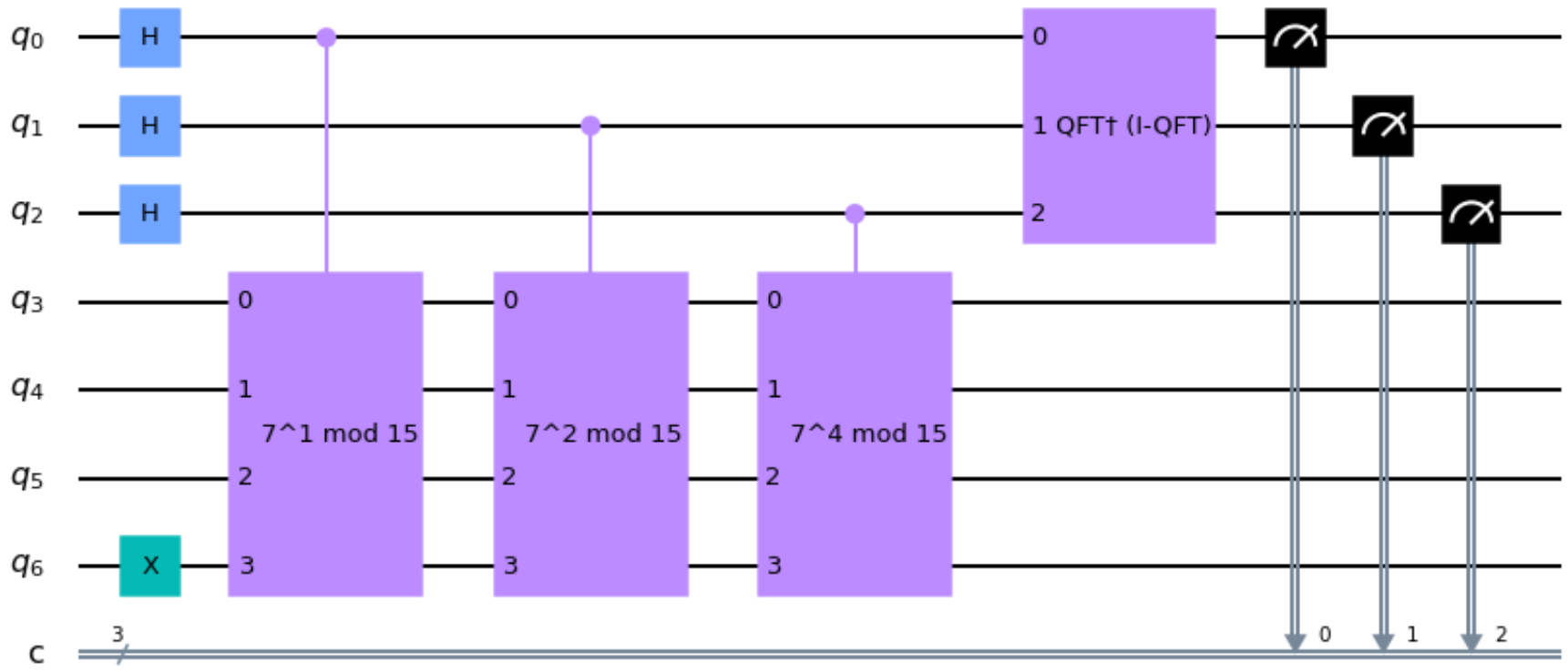
- 함수 $f(x) = a^x \pmod N$의 양자 위상 추정(QPE)

```
def qpe_amod15(a):
    n_count = 3
    qc = QuantumCircuit(4 + n_count, n_count)
    for q in range(n_count):
        qc.h(q)
    qc.x(3 + n_count)
    for q in range(n_count):
        qc.append(c_amod15(a, 2 ** q), [q] + [i + n_count for i in range(4)])
    qc.append(qft_dagger(n_count), range(n_count))
    qc.measure(range(n_count), range(n_count))
    backend = Aer.get_backend('qasm_simulator')
    result = execute(qc, backend, shots=1, memory=True).result()
    readings = result.get_memory()
    phase = int(readings[0], 2) / (2 ** n_count)
    return phase, qc
```

9. 쇼어 알고리즘의 양자회로 구현

주니온TV@Youtube
자세히 보면 유익한 코딩 채널

- 양자 위상 추정(QPE)을 위한 양자 회로



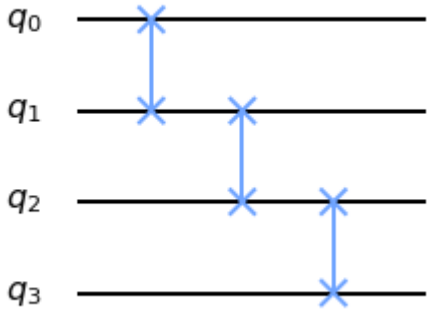
9. 쇼어 알고리즘의 양자회로 구현

- **모듈러 거듭제곱 연산을 위한 양자 회로**

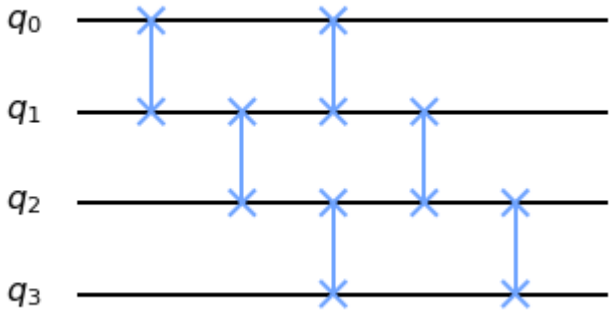
```
def c_amod15(a, power):
    U = QuantumCircuit(4)
    for iteration in range(power):
        if a in [2, 13]:
            U.swap(0, 1); U.swap(1, 2); U.swap(2, 3)
        if a in [7, 8]:
            U.swap(2, 3); U.swap(1, 2); U.swap(0, 1)
        if a == 11:
            U.swap(1, 3); U.swap(0, 2)
        if a in [7, 11, 13]:
            for q in range(4):
                U.x(q)
    U = U.to_gate()
    U.name = " %i^%i mod 15" % (a, power)
    c_U = U.control()
    return c_U
```

9. 쇼어 알고리즘의 양자회로 구현

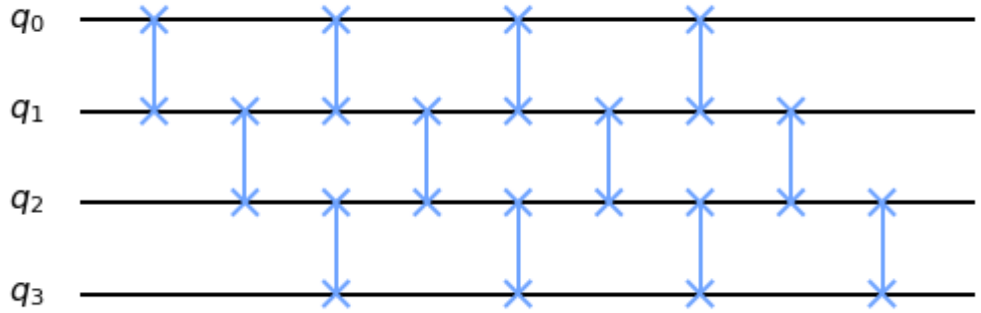
- `c_amod15(a=2, power=2**0)`



- `c_amod15(a=2, power=2**1)`



- `c_amod15(a=2, power=2**2)`



9. 쇼어 알고리즘의 양자회로 구현

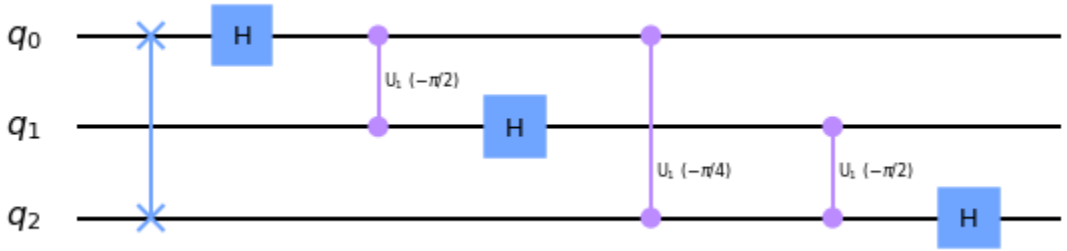
- **양자 역 퓨리에 변환**: Inverse QFT

```
def qft_dagger(n):
    qc = QuantumCircuit(n)
    for qubit in range(n // 2):
        qc.swap(qubit, n - qubit - 1)
    for j in range(n):
        for m in range(j):
            qc.cu1(-np.pi / float(2**(j-m)), m, j)
        qc.h(j)
    qc.name = " QFT† (I-QFT)"
    return qc
```

9. 쇼어 알고리즘의 양자회로 구현

- I-QFT 양자 회로

  - `qft_dagger(3)`



9. 쇼어 알고리즘의 양자회로 구현

- 가슴 벅찬 감동의 순간! 나만 감동인가?

```
def main():
    N = 3 * 5
    p, q, qc = factorize4(N)
    print(N, '=', p, '*', q)
    qc.draw()

main()
```

9. 쇼어 알고리즘의 양자회로 구현

- Qiskit **설치**
  - pip install qiskit

- Qiskit **문서 주소**
  - https://qiskit.org/textbook/ch-algorithms/shor.html

- IBM Q Experience
  - https://quantum-computing.ibm.com/

주니온TV@Youtube

자세히 보면 유익한 코딩 채널

https://bit.ly/2JXXGqz

- 여러분의 구독과 좋아요는 강의제작에 큰 힘이 됩니다.
- 강의자료 및 소스코드: 구글 드라이브에서 다운로드
  (다운로드 주소는 영상 하단 설명란 참고)

  https://bit.ly/3baJZBx