Exercises for Introduction to Quantum Computing

Name: Pugazharasu Anancia Devaneyan (s6puanan)

Matriculation number: 3300280

In [7]:
```
import numpy as np
import sympy
import sys
import fractions
import random
import qiskit
import qiskit.quantum_info as qi
import qiskit.quantum_info as quantu_info
import math
import matplotlib as plt
from qiskit import Aer
from qiskit.utils import QuantumInstance
from qiskit import IBMQ
from qiskit.providers.ibmq import least_busy
from qiskit import QuantumCircuit, transpile, execute
from qiskit.visualization import plot_histogram
from qiskit.circuit.library import QFT
from qiskit.providers.aer import Aer
```

1. Shor's algorithm

We have the unitary,

$$U_{13}|y\rangle = |13y \mod 15\rangle \quad \text{for} \quad y < 15,$$
$$U_{13}|15\rangle = |15\rangle$$

to compute it's eigenvalues and eigenvectors, we use the formula from the lecture,

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{h=0}^{r-1} \exp\left(\frac{-2\pi ish}{r}\right) \left|x^h(\bmod N)\right\rangle, \quad 0 \leq s \leq r-1$$

where $|u_s\rangle$ is an eigenvector of U_x with the eigenvalue,

$$\exp\left(\frac{2\pi is}{r}\right)$$

Thus, for $x = 13$, we have the eigenvalues,

$$\{1, i, -1, -i\}$$

and their respective eigenvectors

$$\frac{1}{2}(|1\rangle + |13\rangle + |4\rangle + |7\rangle)$$

$$\frac{1}{2}(|1\rangle - i|13\rangle - |4\rangle + i|7\rangle)$$

$$\frac{1}{2}(|1\rangle - i|13\rangle|4\rangle - i|7\rangle)$$

Now to find $S \subset N$ such that,

$$\frac{1}{\sqrt{N}} \sum_{i \in S} |i\rangle = |12\rangle$$

where $|i\rangle \in S$. Looking at the eigenvectors we can see that the set S is an improper subset of N, that is it is N itself as summing all the elements in N results in $|12\rangle$

2. Breaking RSA

a) We have the key pair,

$$\{e, N\} = \{20579, 121130231\}$$

We will use Shor's algorithm to factor N. For that first we pick a random number $1 < a < N$, and compute

$$K = gcd(a, N)$$

For this we will use Euclid's algorithm.

In [19]:

```
def gcd(m,n):
    """Obtains the greatest common divisor between two numbers using Euclid

    Args:
        m (int): Number 1
        n (int): Number 2

    Returns:
        int : Returns the greatest common divisor between the two numbers t
    """
    if m< n:
        (m,n) = (n,m)
    if(m%n) == 0:
        return n
    else:
        return (gcd(n, m % n)) # recursion taking place

#setting N
N = 121130231

#Picking a random number 1 < a < N
np.random.seed(1)
a = random.randint(2, N)

#Outputting the results
print(a)

print(gcd(a,N))
```

```
4469197
1
```

Since the GCD is 1 between the random number and N, we would need to find the period then, we do this by means of a classical algorithm

In [20]:

```
def find_period_classical(a, N):
    """Finds the period r of a^r mod N by means of brute forcing

    Args:
        a (int): the number we raise to the power
        N (int): the number we modulo by

    Returns:
        int : Returns the period r
    """
    r = 1
    t = a
    while t != 1:
        t *= a
        t %= N
        r += 1
    return r

r = find_period_classical(a, N)

print(r)
```

```
12110700
```

Now let's check if the period r is even,

In [21]:
```
print(r%2)
```

```
0
```

We see that r is indeed even, we can now use the formula,

$$p = gcd(a^{\frac{r}{2}} + 1, N)$$

and

$$q = \frac{N}{p}$$

to compute the prime facors of N.

In [22]:
```
a = 4469197
N = 121130231
r = 12110700

p = math.gcd(a**int(r/2)+1,N)
q = math.gcd(a**int(r/2)-1,N)
print(p,q)
```

```
7901 15331
```

We find that,

$$N = pq$$

where $q = 15331$ and $p = 7901$.

b) Based on the example, the string 'blhhay' maps to the number,

$$blhhay = (1 \times 26^5) + (11 \times 26^4) + (7 \times 26^3) + (7 \times 26^2) + (0 \times 26) + 24 = 17035900$$

We can decrypt this by first figuring out the private key d. This we can do since we found from (a) the prime factors of the public key, thus we compute the multiplicate inverse of $e =$, to find that

$$d = 20579$$

We can thus now decrypt the message by applying

$$M = E^d \bmod N = 17035900^{20579} \bmod 121130231$$

we compute this to find that,

$$M = 40574$$

Now that we have the decrypted message, we simply need to express it in a base-26 number system in order to obtain the message in terms of alphabets

In [6]:

```
#Decrypted message as a number
xx = 40574

#Decoding the message by representing it using a base-26 number system
x1 = int(xx//(26**5))
factor_1 = x1 * (26**5)
x2 = int((xx - factor_1)//(26**4))
factor_2 = x2 * (26**4)
x3 = int((xx - factor_1 - factor_2)//(26**3))
factor_3 = x3 * (26**3)
x4 = int((xx - factor_1 - factor_2 - factor_3)//(26**2))
factor_4 = x4 * (26**2)
x5 = int((xx - factor_1 - factor_2 - factor_3 - factor_4)//(26**1))
factor_5 = x5 * (26)
x6 = int(xx - factor_1 - factor_2 - factor_3 - factor_4 - factor_5)
display(x1,x2,x3,x4,x5,x6)
```

```
0
0
2
8
0
14
```

Using the cipher given the exercise sheet we find that the decrypted message is "aaciao"