Shor's Algorithm

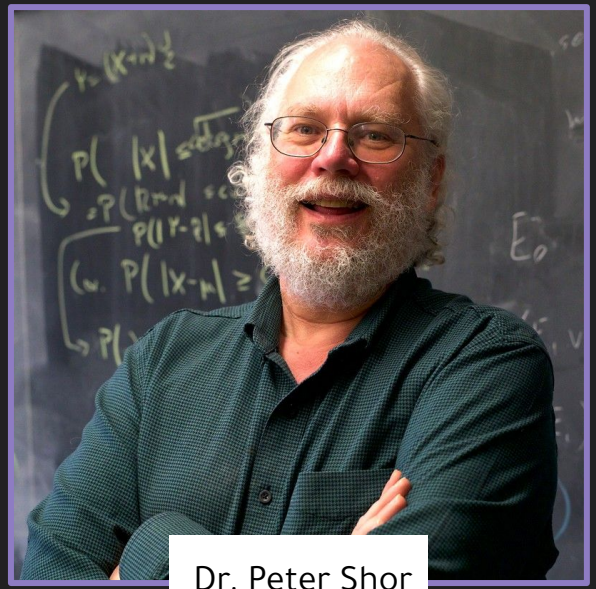
Stephanie Long

NYU Shabani Lab Quanum Internship

Shor’s Algorithm

Who is Shor

- Dr. Shor is an applied mathematics professor at MIT
- Expert in quantum computation
- Discovered Shor's Algorithm in 1994



Dr. Peter Shor

What is an algorithm?

Mathematics Definition

A procedure, a set of steps that can be used to solve a mathematical computation

Engineering Definition

A sequence of well-defined steps that produce a result in finite time

Classical CS Definition

A specific procedure for solving a well-defined computational problem

Quantum CS Definition

A procedure to solve a problem only functional with a quantum computer

Problem

It is not possible to factor a number into it's prime factors in reasonable time using a classical computer (sub-exponential time)

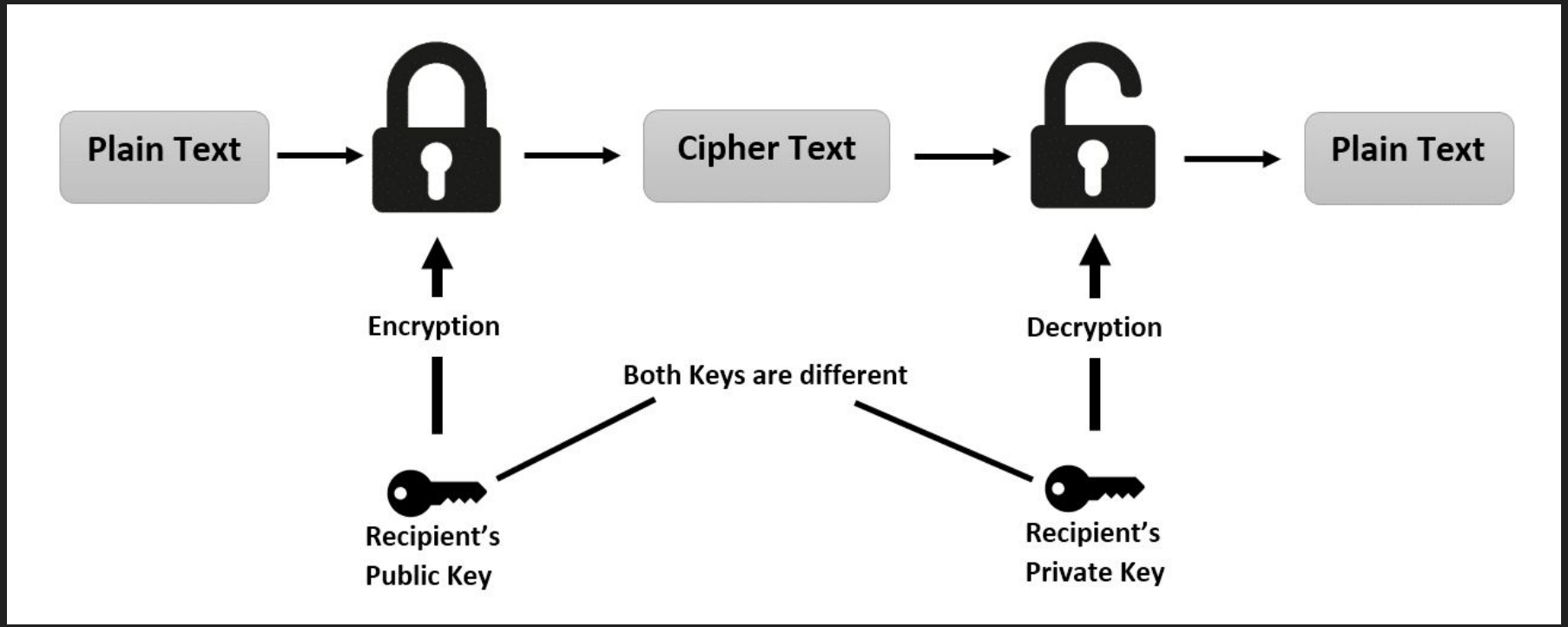
Solution

Quantum computers can factor an extremely large number into it's prime factors in reasonable time (polynomial time)

Useful for decrypting the RSA cryptosystem

What is RSA cryptography?

RSA is an encryption method that we use everyday - for secure emails, transactions, and more



More Details : RSA

RSA is based off of extremely large numbers - takes **1,024**, **2,048**, or **4,096** bits to represent

$2^n-1= \#$

n = bits

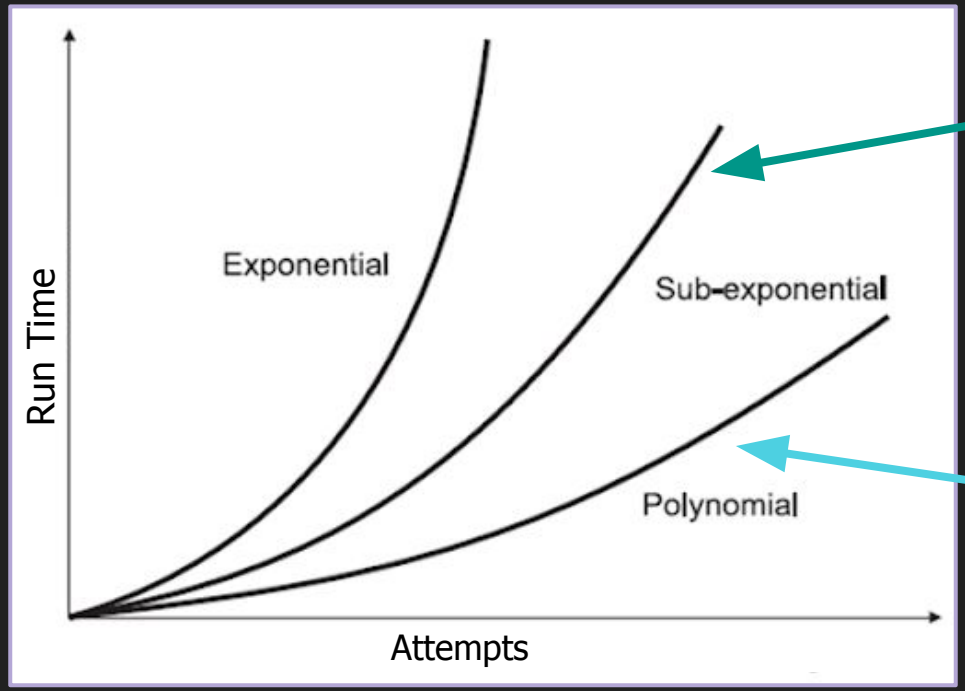
→ Number is too large - overflow error

A key is essentially a prime factor of the RSA number. If you have the RSA number and a prime factor you can deduce the other prime factor and the plain text.

The encryption operates based on the presumption that a computer cannot factor such a large number into it's primes.

More Details : RSA

The algorithm being used on classical computers to factor extremely large numbers today is the general field number sieve



Classical Computer factoring (GFNS)

Quantum Computer factoring (Shor's Algorithm)

Shor's Algorithm - CLASSICAL IMPLEMENTATION

N = extremely large number

STEP 1)

- Generate a random number "a" which satisfies the condition 0 < a < N-1

```
N = ##insert number here##
a = randint(2,N-1)
print("A random number a has been generated:  " + str(a))
```

Shor's Algorithm

N = extremely large number

STEP 2)

- Compute the greatest common denominator of a & N (using Euclid's Algorithm)
  - If GCD is != a nontrivial factor has been found, return to step 1

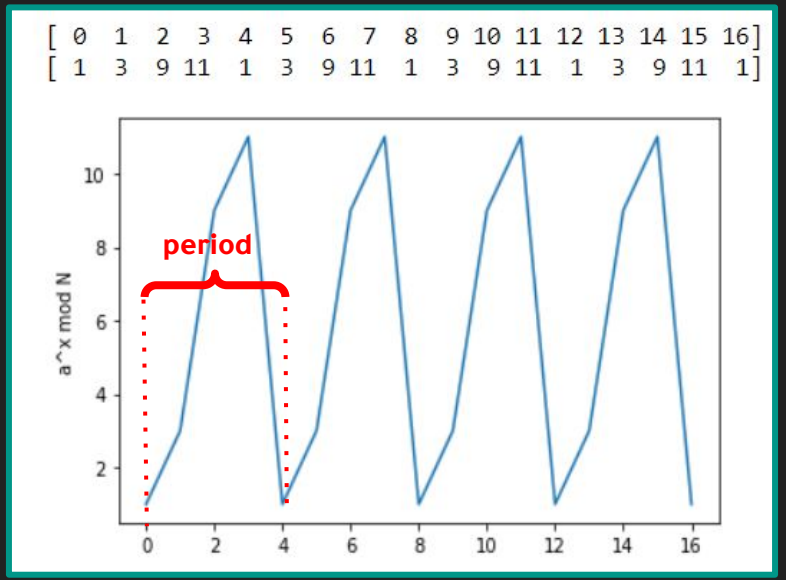
```
def gcd(pick, number):
    if pick == 0 :
        return number

    return gcd(number%pick, pick)

print("The greatest common denominator of (",a,",",N, ") is", gcd(a, N))
```

STEP 3)

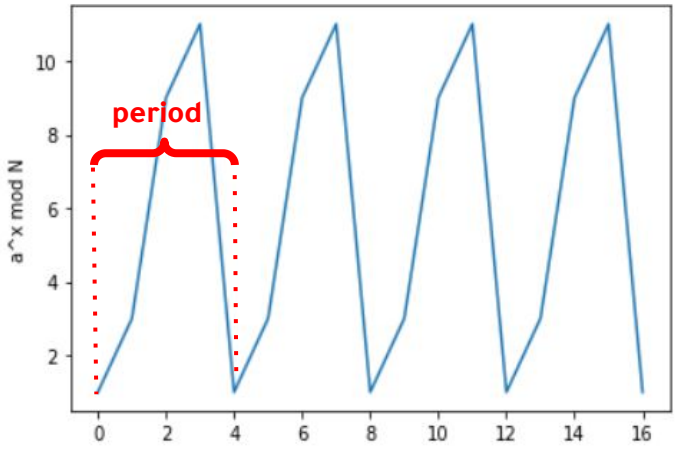
- Graph the equation $y = a^x \bmod N$ from x value 0 to (N-1)



STEP 4)

- Find the period using the equation $a^r \bmod N = 1$

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16]
[ 1  3  9 11  1  3  9 11  1  3  9 11  1  3  9 11  1]
```

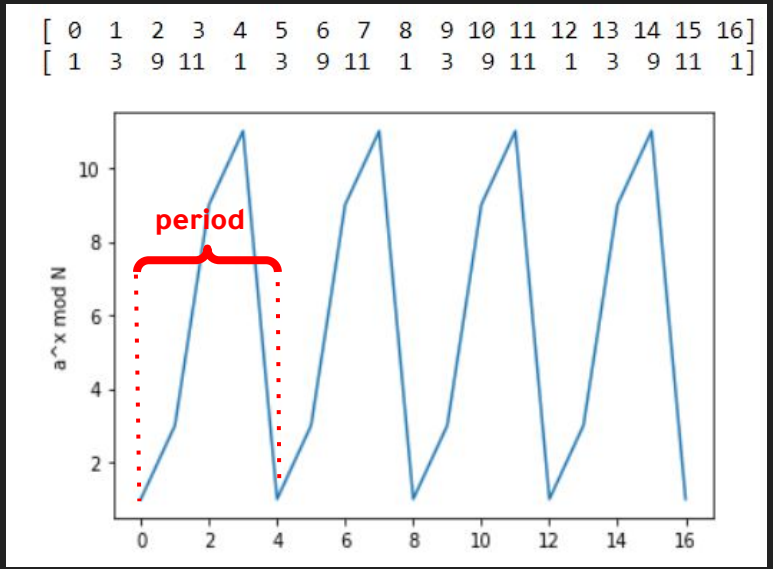


STEP 5)

- Run the period (r) through a series of checks

STEP 6)

- If r passes all checks, then you are more likely to have found the factors, which are found when plugged into the below equations

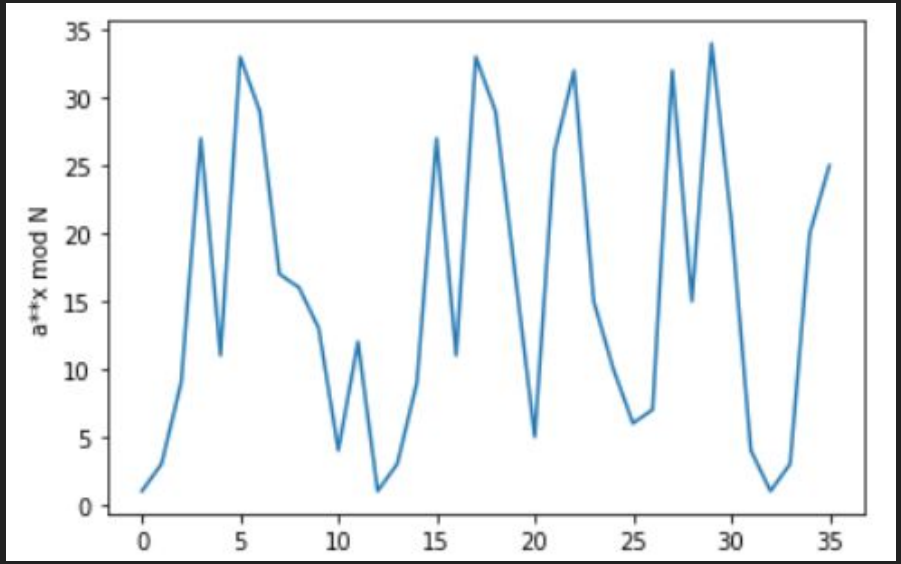


$$gcd(a^r/2 + 1)$$
$$gcd(a^r/2 - 1)$$

So, what's the issue here?

- The N value in the period finding was 16 - a small number

- Classical computation is not fast enough for Shor's Algorithm



--N is increased--

Error occurs - no viable period

Quantum Circuit Implementation

- By implementing the use of a quantum algorithm in the period finding step, we can speed up the algorithm and more efficiently factor large numbers

Why does speed up occur?

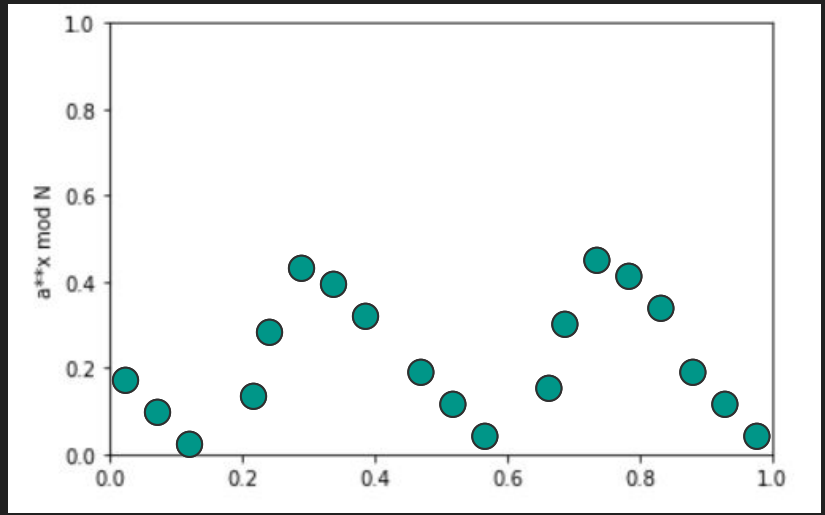
- Classical computers can only find

  one x value at a time

- Due to superposition, quantum

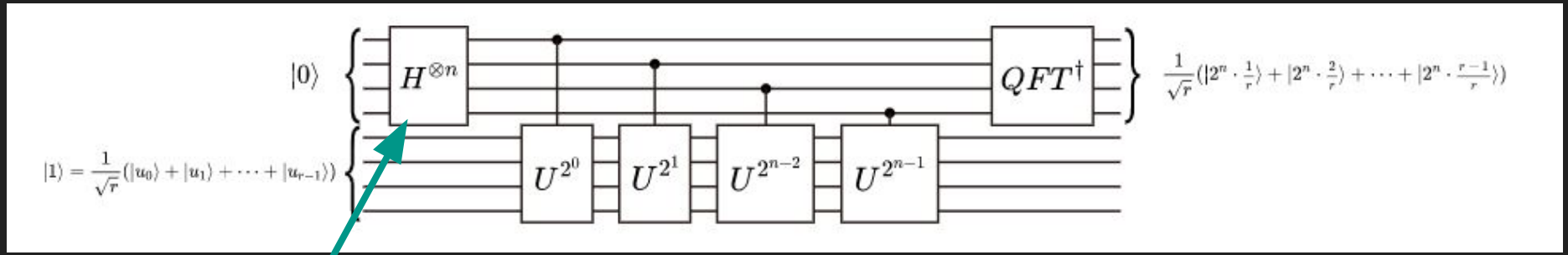
  computers can find multiple x values

  at a time



Quantum Circuit Implementation

Lets take a look at the details:



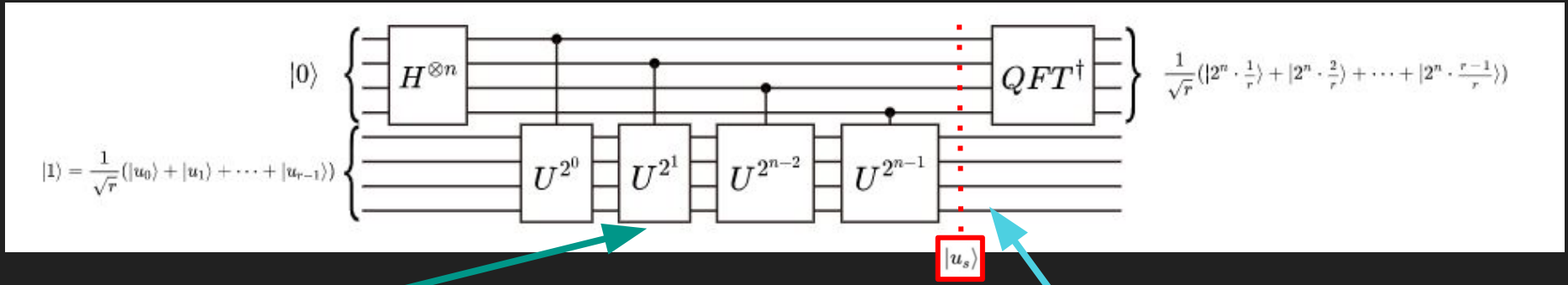
$$H = 1/\sqrt{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

- Hadamard gate puts the qubits into a superposition of states
- This superposition allows more than one x-value to be calculated at once - in opposition to the limited calculations of classical bits

$$(|0\rangle + |1\rangle)/\sqrt{2} \rightarrow |\psi\rangle = (|00\rangle + |10\rangle + |01\rangle + |11\rangle)/2$$

Quantum Circuit Implementation

Lets take a look at the details:



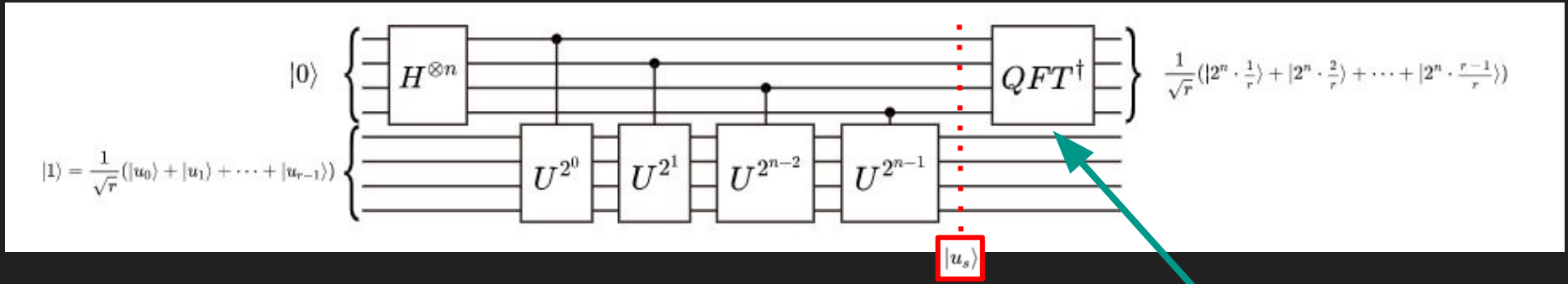
Unitary Gates :

Perform rotations on the qubits

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-\frac{2\pi i s k}{r}} |a^k \bmod N\rangle$$

Quantum Circuit Implementation

Lets take a look at the details:



- Produces final superposition
- The period can be extrapolated from this information

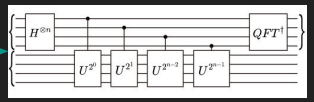
Quantum Fourier Transform:

Acts upon the state generated by the unitary gates

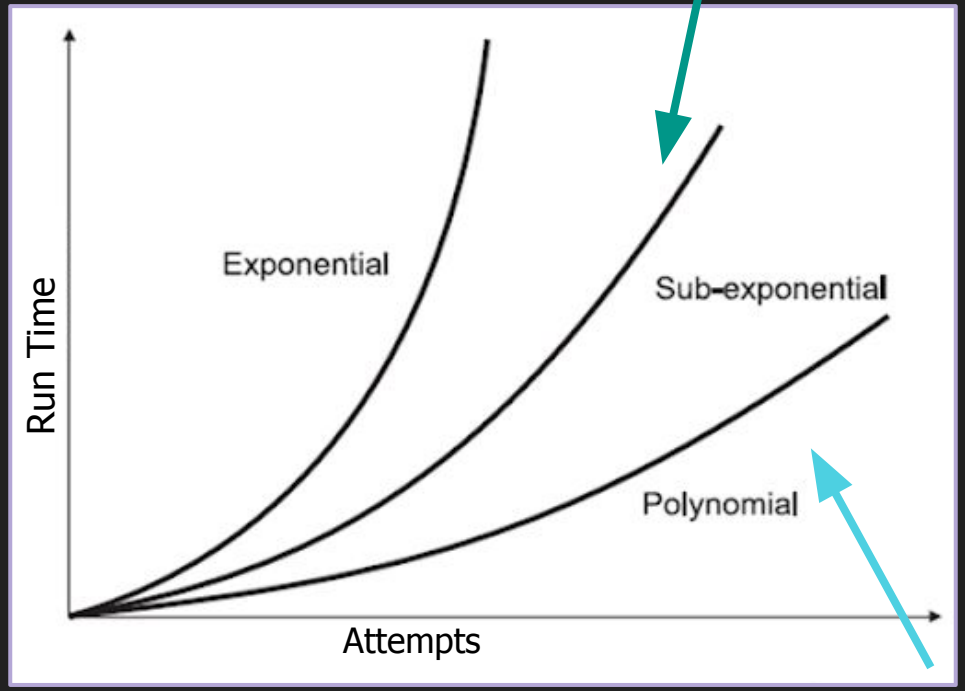
Reminder : Why is Quantum Helpful?

By exchanging the Classical method with the Quantum method of period finding

$$a^r \bmod N = 1$$



the time of computation is reduced from sub-exponential time to polynomial time, making the algorithm possible to run



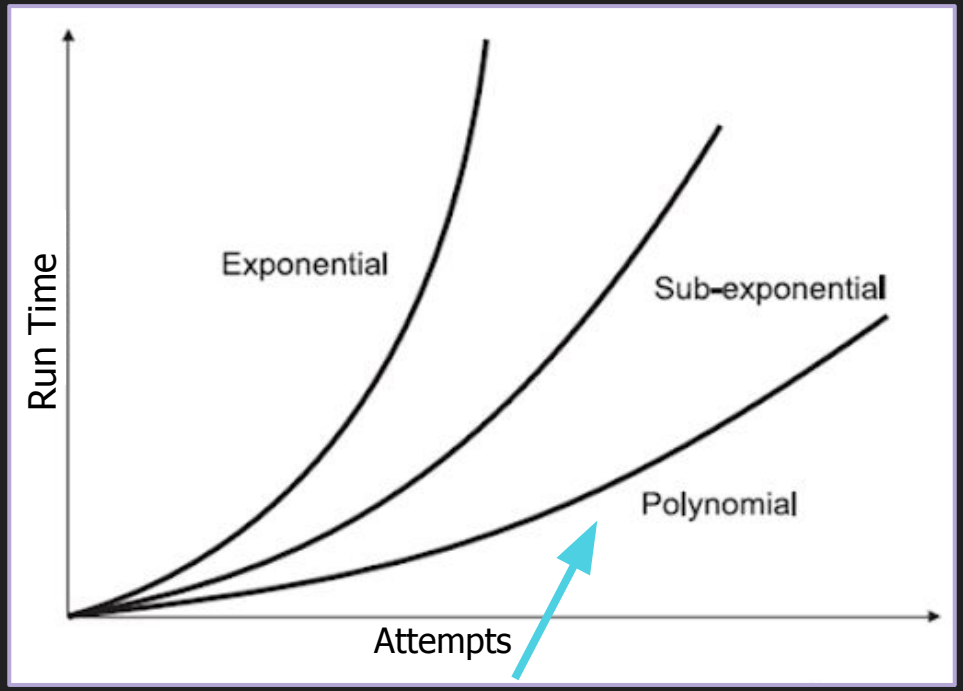
How fast is Shor's Algorithm

- Compared to classical, it is an "exponential speed up"
- y^{10} seconds v.s. 10^{y} seconds, y = digits

quantum

classical

- EX. to break a 2048 bit encryption would take a working Quantum Computer 8 hours, and a classical computer 300 trillion years



Quantum Computer factoring (Shor's Algorithm)

References

1) https://math.mit.edu/directory/profile.php?pid=247
2) https://www.quantum-inspire.com/kbase/what-is-a-quantum-algorithm/
3) https://www.youtube.com/watch?v=ufO_BScIHDQ
4) https://www.geeksforgeeks.org/euclidean-algorithms-basic-and-extended/
5) https://en.wikipedia.org/wiki/Quantum_algorithm
6) https://en.wikipedia.org/wiki/General_number_field_sieve
7) https://www.quantiki.org/wiki/shors-factoring-algorithm#:~:text=Shor's%20algorithm%20is%20a%20quantum,a%20sufficiently%20large%20quantum%20computer.
8) https://quantum-computing.ibm.com/composer/docs/iqx/guide/shors-algorithm
9) https://matplotlib.org/stable/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py
10) https://www.w3schools.com/python/python_functions.asp
11) https://numpy.org/doc/stable/reference/generated/numpy.fromfunction.html
12) https://www.w3schools.com/python/python_conditions.asp
13) https://en.wikipedia.org/wiki/RSA_(cryptosystem)
14) https://qiskit.org/textbook/ch-algorithms/shor.html
15) https://en.wikipedia.org/wiki/Shor%27s_algorithm
16) https://news.mit.edu/2016/quantum-computer-end-encryption-schemes-0303
17) https://www.protocol.com/manuals/quantum-computing/quantum-computers-wont-break-encryption-yet#toggle-gdpr
18) https://www.quintessencelabs.com/blog/breaking-rsa-encryption-update-state-art/#:~:text=It%20would%20take%20a%20classical,RSA%2D2048%20bit%20encryption%20key.

Shor's Algorithm

Stephanie Long