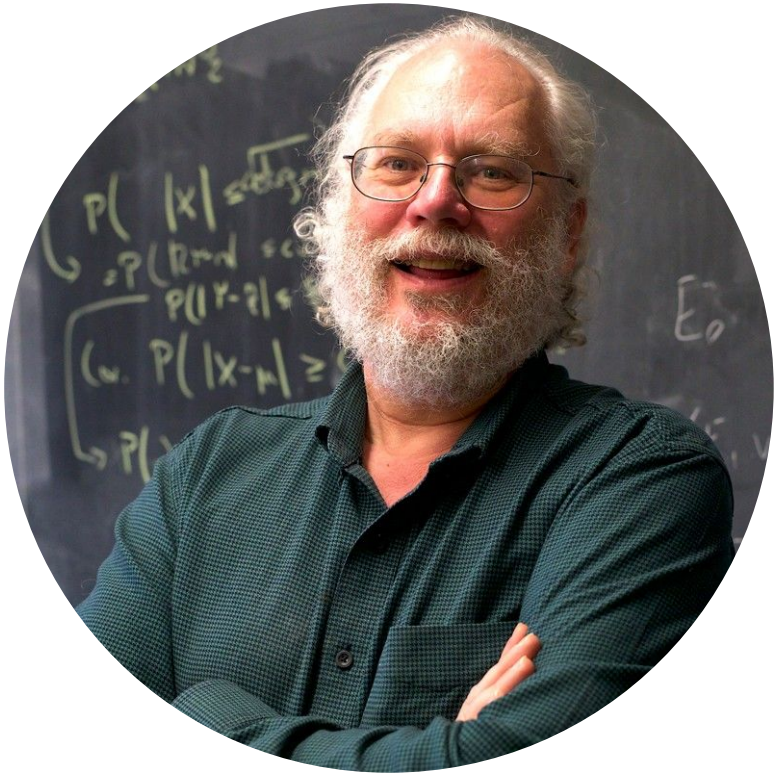


Shor's Factoring Algorithm

Sharon Zlatnik~ Forest Hills High School (AEOP)
Isabela Chaveztello~ Riverdale Country School
Ariam Yohannes~ Westhill High School

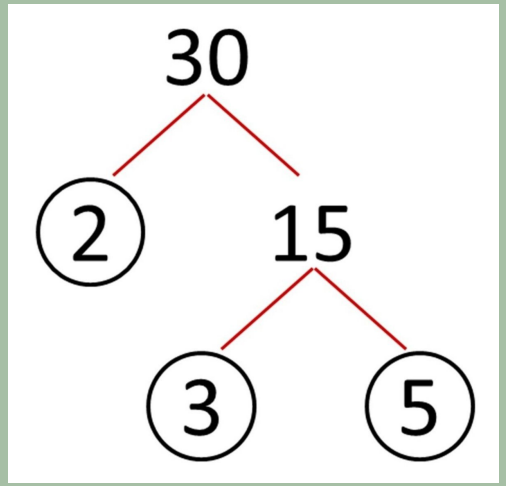
Table of contents

Important definitions

I. **Prime number**
   - A number that cannot be formed by multiplying two smaller numbers
   - Only two distinct divisors: 1 and itself

II. **Factoring**
   - Factoring a number into its prime factors is expressing the number as a product of prime numbers.

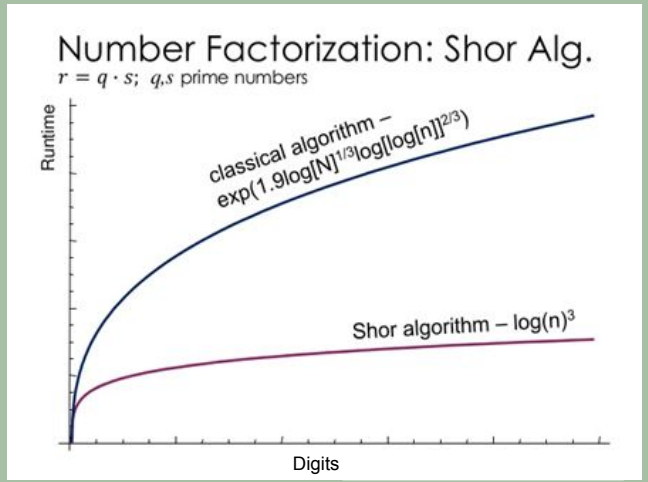
III. **Co-prime numbers**
   - Two numbers that have no common positive divisors other than 1.
   - GCD(g, N) = 1



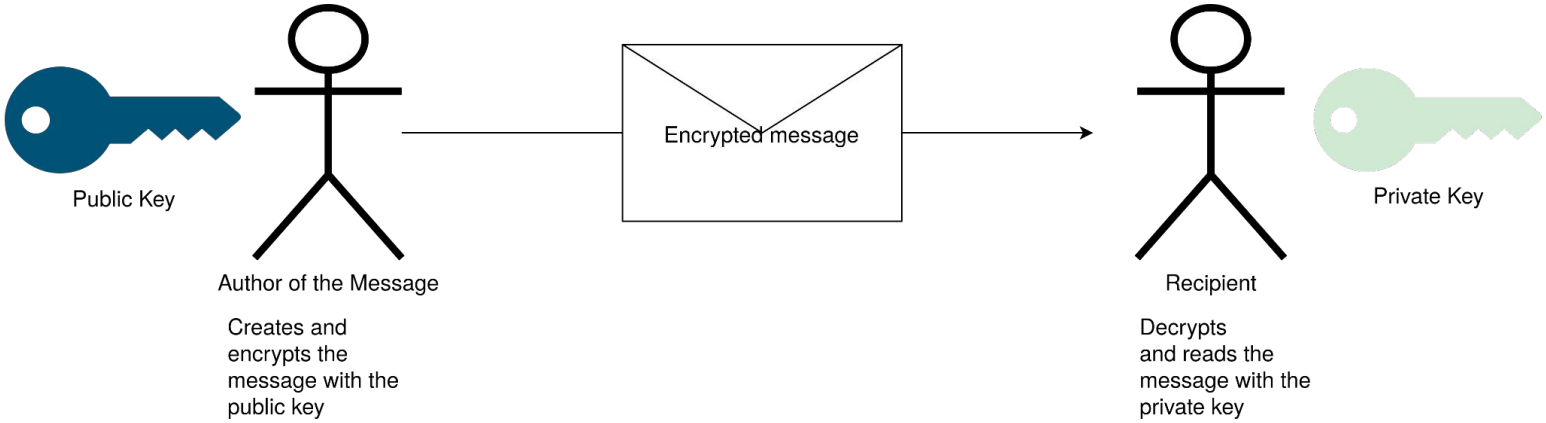
What is Shor's algorithm?

I. **What is it?**
   - Factorization algorithm
   - Splits integers into prime factors
   - Classical-Quantum Hybrid

II. **Why do we care?**
   - Logarithmic runtime v polynomial runtime
   - Applicable to real-world problems
     - Ex: RSA cryptography



RSA and factoring



RSA- 2048

```
25195908475657893494027183240048398571429282126204032027777137836043662020707595556264018525880784406918290641249515082189298559149176184502808489120072844992687392807287776735971418347270261896375014971824691165077613379859095700097330459748808428401797429100642458691817195118746121515172654632288216869987549182422433637259085141865462043576798423387184774447920739934236584823824281198163815010674810451660377306056201619676256133844143603833904414952634432190114657544454178424020924616515723350778707749817125772467962926386356373289912154831438167899885040445364023527381951378636564391212010397122822120720357
```

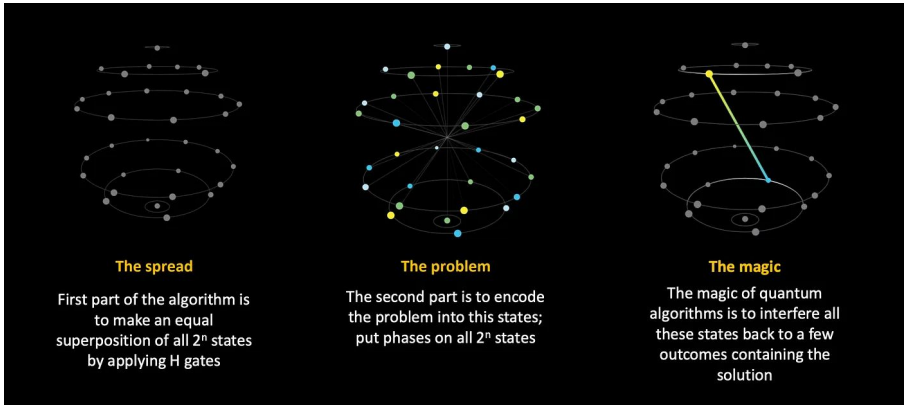
Classical

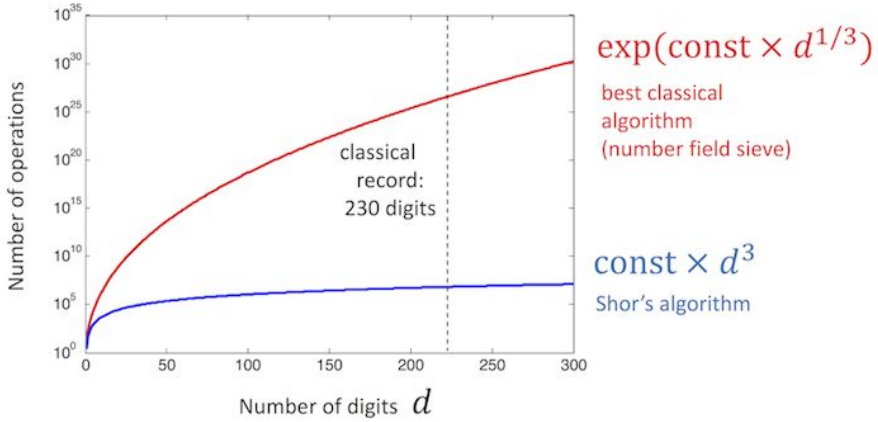
- On classical computers Shor's Algorithm would not be the fastest way to factorize integers
- Loop through 0 - √num.
- Add number to answers if the number is divisible (num % generated num = 0)

```
algorithm Sieve of Eratosthenes is
    input: an integer n > 1.
    output: all prime numbers from 2 through n.

    let A be an array of Boolean values, indexed by integers 2 to n,
    initially all set to true.

    for i = 2, 3, 4, ..., not exceeding √n do
        if A[i] is true
            for j = i², i²+i, i²+2i, i²+3i, ..., not exceeding n do
                set A[j] := false

    return all i such that A[i] is true.
```

Quantum

I. Uses superposition to speed up computation
II. Shor's Algorithm is Classical-Quantum Hybrid





Explaining Shor's Algorithm

1. We will explain how factoring N is related to period determination
2. We will explain how QPE contributes to this and how we utilize QC to create a hybrid Classical-Quantum Algorithm

Make a guess

Ensure that both the guess and the number fulfills the conditions

Find the period

Find the value of r that fulfills $a^{r}=1 \mod N$

Euclid's Algorithm

$21 \div 4 = 5$ remainder 1
$4 \div 1 = 4$ remainder 0

Therefore, GCD(21, 4) is 1

The Math (Number Theory)

I. G is coprime with N and 2<a<N

II. Euler's Theorem

   A. If a,N are coprime there is a least power so that $a^r=1$ modN (r is an order of amodN)

III. Modular Exponential Functions

$$f(r) = a^r mod N$$

$$(A, B) \rightarrow A * A * A * A .. * A = m * B + 1$$

$$A^r = m * B + 1$$

$$a^r = m * N - 1 \mapsto a^r - 1 = m * N$$

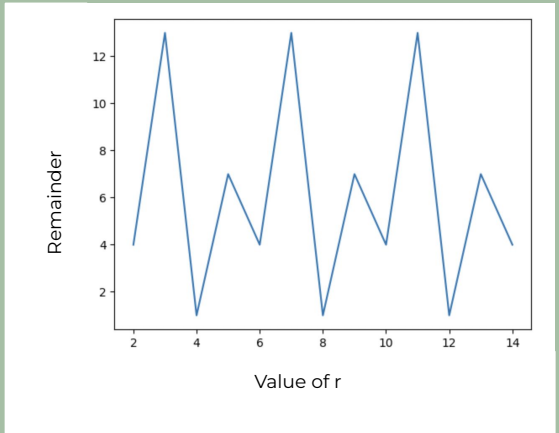
$$a^r - 1 = (a^{r/2} - 1)(a^{r/2} + 1)$$

IV. Euclid's Algorithm

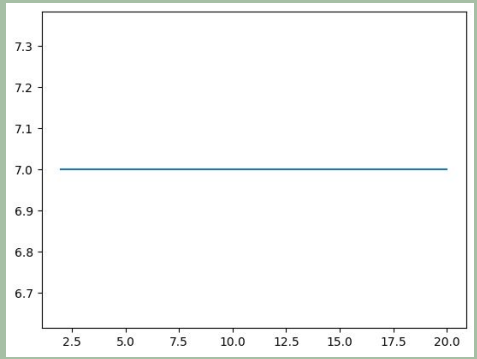
   A. $gcd(a^{r/2} \pm 1, N)$

The period finding problem + conditions

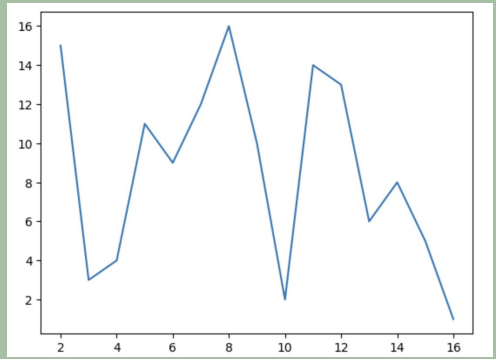
N = 15 (not prime)
a = 7



N = 21 (Not coprime)
a = 7



N = 23 (prime)
a = 7



Make a guess

*** Condition: N, number we are trying to factor cannot be even or prime!

- Not 8 because you know a prime factor is 2, meaning the problem is solved
- Not 17 because only factors are 1 and 17

***Condition: guess, g, is coprime with the number! GCD(N, g) = 1
If not, algorithm is complete

*** Condition: guess, g is within these parameters: $2 < g < N$

Assume our number N = 15

Our guess is 7

GCD(15, 7) = 1 ← fulfills the condition

Find period

g is our guess, 7
r is the value we are trying to find (period)
N is the number we are trying to factor
$7^r \bmod 15 = 1$ ←this is what we are trying to do ($g^r \bmod N = 1$), solve for r

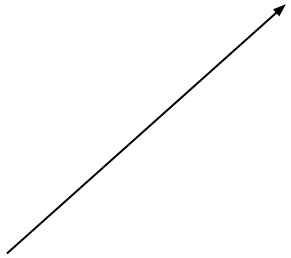
Find period $g^r \bmod N = 1$

$1 * 7 \% 15 = 7$ ...1

$7 * 7 \% 15 = 4$ ...2

$4 * 7 \% 15 = 13$ ...3

$13 * 7 \% 15 = 1$ ...4



Period, r, = 4

Repeat these steps until you get 1 as a remainder. The amount of steps it took was the period. This is the classical way to compute period, but this step takes very long with large numbers. Quantum computers, however, can complete this step very quickly

*** Condition: If r is odd, redo with a different guess!!!

Perfecting our Guess

To perfect our guess, we must find the value of $g^{r/2} \bmod N + 1$ and $g^{r/2} \bmod N - 1$.

We have r = 4, g = 7, N = 15

So to find the correct guess, you must do $7^{4/2} \bmod 15 + 1$ and $7^{4/2} \bmod 15 - 1$.
$7^2 \bmod 15 + 1$ and $7^2 \bmod 15 - 1$.

5 3

***Condition: f you get N - 1 as the answer to these equations, you must redo with a different guess

Use Euclid's algorithm for final answer

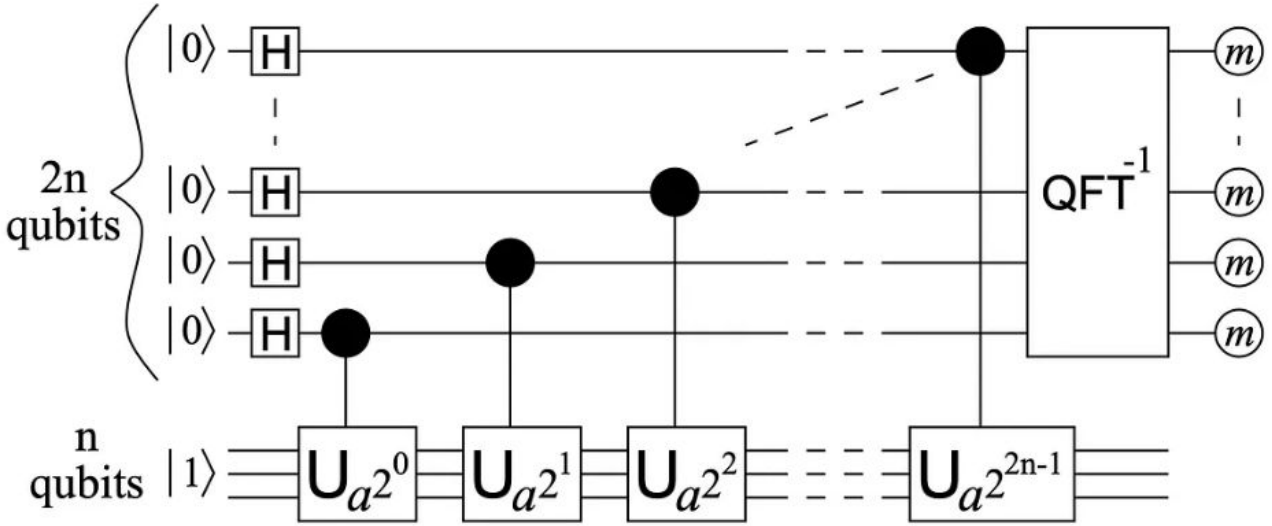
Euclid's algorithm is used to find GCD between two numbers.
Example:
123 / 36 = 3 remainder 15
36 / 15 = 2 remainder 6
15 / 6 = 2 remainder 3
6 / 3 = 2 remainder 0

GCD(123, 36) = 3

With our numbers:
15 / 5 = 0
GCD(15, 5) = 5
15 / 3 = 0
GCD(15, 3) = 3

So our final answers are 3 and 5! As long as you follow the preconditions, you should be able to derive the factors of any number using Shor's algorithm with a 60-80% success rate

Period Finding Circuit



Order Finding Algorithm

https://medium.com/mit-6-s089-intro-to-quantum-computing/a-general-implementation-of-shors-algorithm-da1595694430

Modular Multiplication Circuit

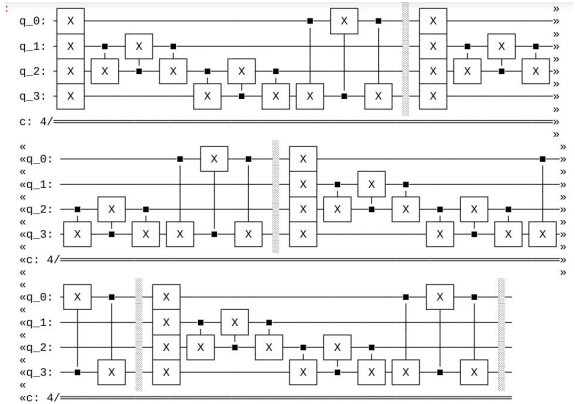
Unitary Gates

$$f(x) = ax \pmod{N}$$

$$U|y\rangle = |xy \mod N\rangle$$

```
## Modular Multiplication Circuit
n = 4
circ = QuantumCircuit(n,n)

def unitary_7(circuit):
    for i in range(n):
        circuit.x(i)
    circuit = swap(circuit,1,2)
    circuit = swap(circuit,2,3)
    circuit = swap(circuit,0,3)
    circuit.barrier()
    return circuit


def swap(circuit, i, j):
    circuit.cx(i,j)
    circuit.cx(j,i)
    circuit.cx(i,j)
    return circuit


r = 4
for j in range(r):
    circ = unitary_7(circ)
circ.draw()
```

a=7, N=15



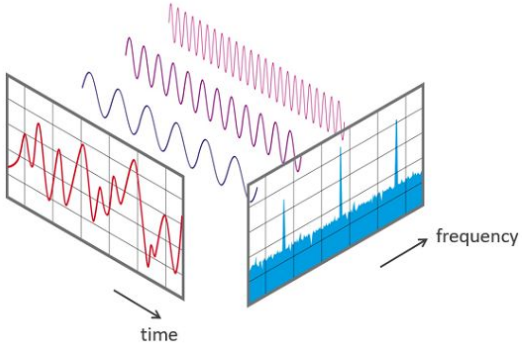
Our code and circuit

period of $|x^a \mod N\rangle = r$

and

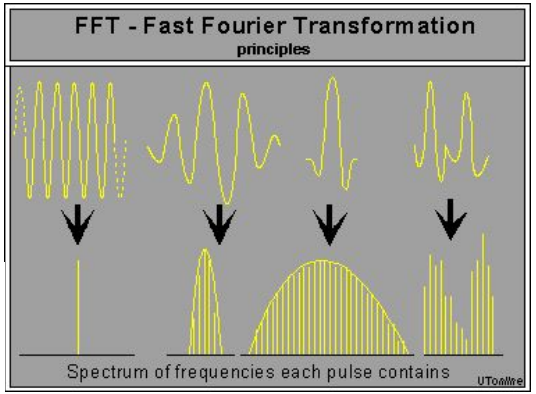
$$U|u_s\rangle = \exp\left[\frac{2\pi i s}{r}\right]|u_s\rangle$$

Fast Fourier Transform



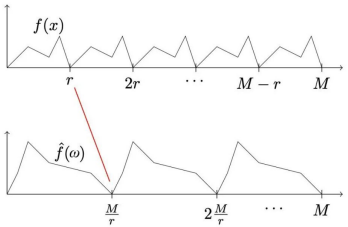
https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft

Time domain Vs Frequency Domain

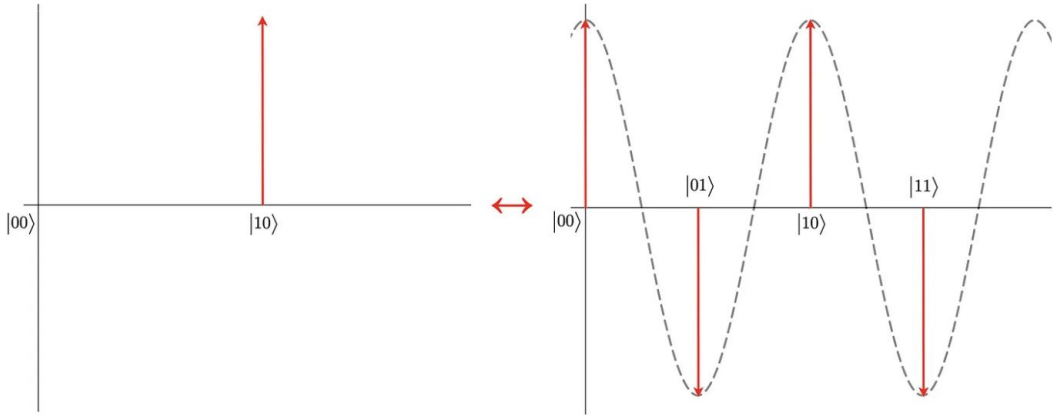


https://www.ndt.net/ndtaz/content.php?id=163

If the Period is R in the time domain than the period will be ½ in the frequency domain



https://jonathan-hui.medium.com/qc-quantum-fourier-transform-45436f90a43



i.e., $|10\rangle$ will transform into the superposition below:

$$|10\rangle \leftrightarrow |00\rangle - |01\rangle + |10\rangle - |11\rangle$$

https://jonathan-hui.medium.com/qc-quantum-fourier-transform-45436f90a43

Quantum Fourier Transform/Phase Estimation

$$A\mathbf{v} = \lambda\mathbf{v}$$

eigenvalue

$$\begin{bmatrix} 1 & -3 & 3 \\ 3 & -5 & 3 \\ 6 & -6 & 4 \end{bmatrix} \begin{bmatrix} 1/2 \\ 1/2 \\ 1 \end{bmatrix} = 4 \begin{bmatrix} 1/2 \\ 1/2 \\ 1 \end{bmatrix}$$

A eigenvector

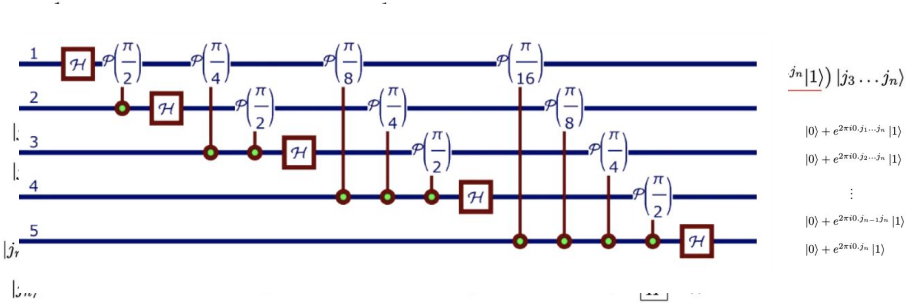
Known Estimate

$$U|\psi\rangle = e^{2\pi i\phi}|\psi\rangle$$

Transforms one quantum state into another encoded with the frequency spectrum of the original state

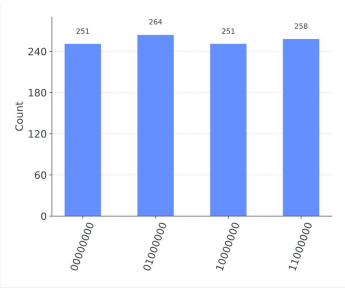
$$\hat{F} = \left(\sum_{j',k'} \frac{e^{2\pi i j'k'/N}}{\sqrt{N}} |k'\rangle\langle j'|\right), \quad \hat{F}^\dagger = \left(\sum_{j,k} \frac{e^{-2\pi i jk/N}}{\sqrt{N}} |j\rangle\langle k|\right)$$

$$\begin{aligned} \hat{F}^\dagger\hat{F} &= \frac{1}{N}\sum_{j,k,j',k'} e^{2\pi i(j'k'-jk)/N}|j\rangle\langle j'|\delta_{kk'} \\ &= \frac{1}{N}\sum_{j,k,j'} e^{2\pi i(j'-j)k/N}|j\rangle\langle j'| \\ &= \sum_{j,j'}|j\rangle\langle j'|\delta_{jj'} = \sum_j |j\rangle\langle j| = \hat{I}. \end{aligned}$$



https://jonathan-hui.medium.com/qc-quantum-fourier-transform-45436f90a43

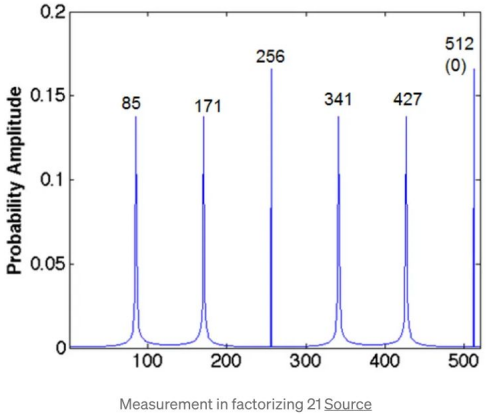
How do we get value for the period?



	Register Output	Phase
0	00000000(bin) = 0(dec)	0/256 = 0.00
1	11000000(bin) = 192(dec)	192/256 = 0.75
2	10000000(bin) = 128(dec)	128/256 = 0.50
3	01000000(bin) = 64(dec)	64/256 = 0.25

	Phase	Fraction	Guess for r
0	0.00	0/1	1
1	0.75	3/4	4
2	0.50	1/2	2
3	0.25	1/4	4

https://learn.qiskit.org/course/ch-algorithms/shors-algorithm



Measurement in factorizing 21 Source

$$\frac{c}{q} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots}}}, \quad d_0 = a_0, \quad d_1 = 1 + a_0 a_1, \quad d_n = a_n d_{n-1} + d_{n-2}$$

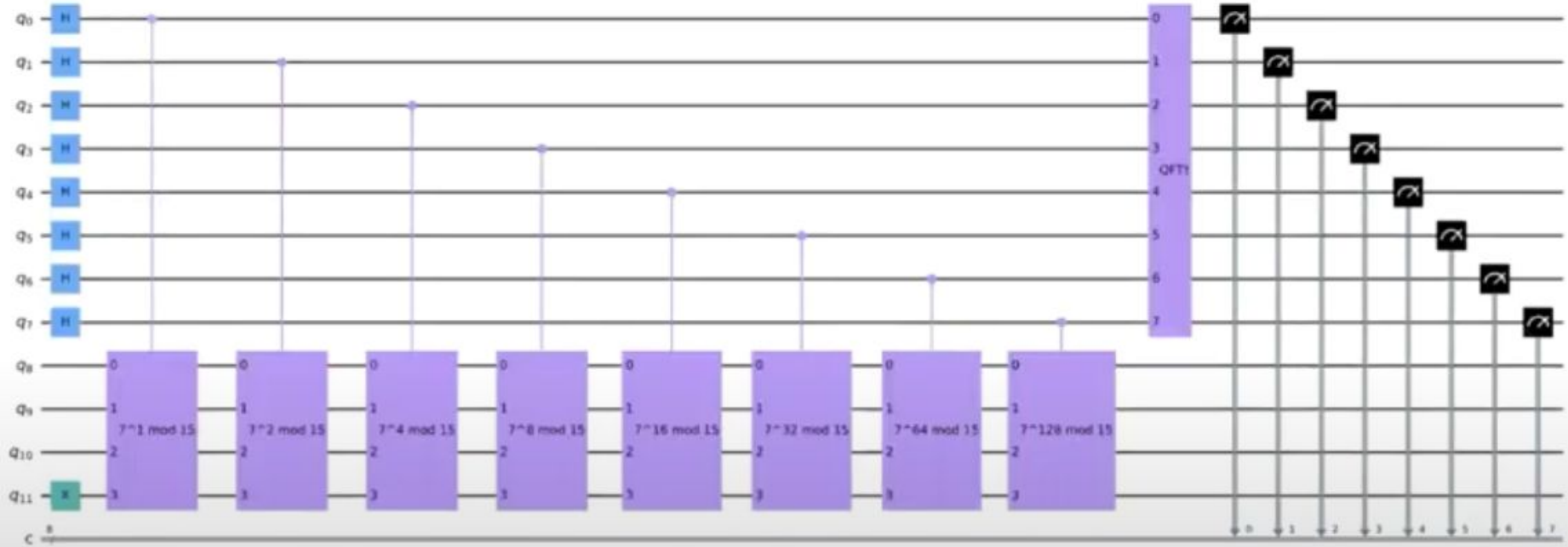
$$r_0 = 1, \quad r_1 = a_1, \quad r_n = a_n r_{n-1} + r_{n-2}$$

$$\frac{427}{512} = 0 + \cfrac{1}{1 + \cfrac{1}{5 + \cfrac{1}{42 + \frac{1}{2}}}}, \quad d_0 = 0, \quad d_1 = 1, \quad d_2 = 5, \quad d_3 = 427$$

$$r_0 = 1, \quad r_1 = 1, \quad r_2 = 6, \quad r_3 = 512$$

https://qudev.phys.ethz.ch/static/content/QSIT15/Shors%20Algorithm.pdf

Qiskit Quantum Circuit for Shor's where g = 7 and N = 15

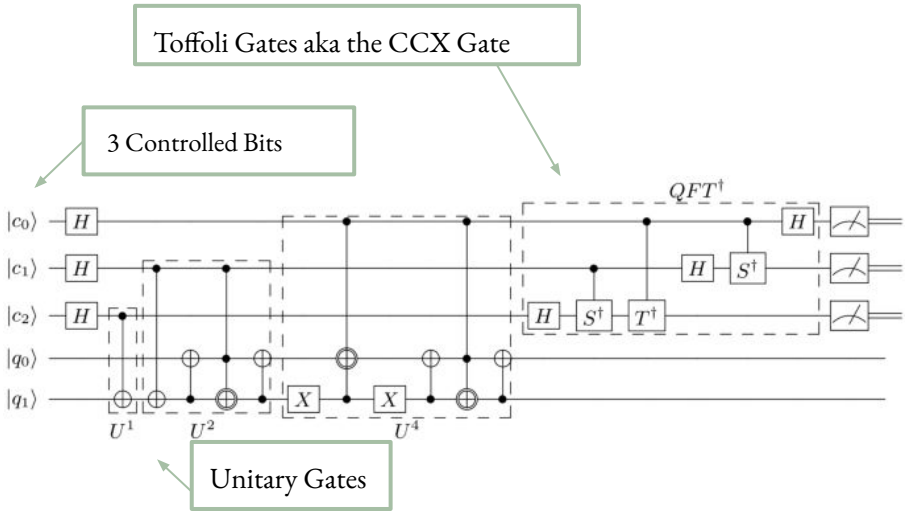


https://learn.qiskit.org/course/ch-algorithms/shors-algorithm

Factorization of 21 on IBM Quantum

Things to Keep in Mind:

I. Complete Implementation of Shor's Algorithm is difficult
II. To maximize efficiency the circuit has to be tailored to the number you are factoring
III. There, as of now, is no universal circuit
IV. Takes a lot of qubits and gates



https://www.nature.com/articles/s41598-021-95973-w

Future

- Posed threat to encryption
- Quantum computing is still in its early stages
- The rise of post-quantum cryptography to protect communication
    - Google, NIST, Microsoft etc

Thank you for listening!

Any questions?

Sources

- https://quantum-computing.ibm.com/composer/docs/iqx/guide/shors-algorithm
- https://www.nature.com/articles/s41598-021-95973-w
-
- https://jonathan-hui.medium.com/qc-phase-estimation-in-shors-algorithm-acef265ebe50#:~:text=Phase%20estimation%20simply%20means%20estimate,phase%20value%20of%20the%20eigenvalue.
- https://medium.com/mit-6-s089-intro-to-quantum-computing/a-general-implementation-of-shors-algorithm-da1595694430
- https://www.quantiki.org/wiki/shors-factoring-algorithm#:~:text=Shor's%20algorithm%20is%20a%20quantum,a%20sufficiently%20large%20quantum%20computer.
- https://scottaaronson.blog/?p=208