Quantum Computing with Grover and Advanced Algorithms

→ Basic Classical computer gates :-

1) Classical Gates
   - AND
   - OR
   - NOT
   - XOR
   - BUFFER (Identity or Nothing gate)

XOR

A	B	A⊕B
0	0	1
0	1	0
1	0	0
1	1	1

- NAND (NOT + AND) ② ①
- NOR (OR + NOT)
- XOR (Exclusive OR)

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

$\overline{XOR}$

- XNOR

Same as XOR

No.
Date

Quantum Gates

- NOT = $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ ✱ Complement of Identity

Some of the gates that we can see in IBM Composer

Hadamard Gate (H)

Creates an equal superposition of $|0\rangle$ and $|1\rangle$.

Effect: If applied to $|0\rangle$,

$$(|0\rangle + |1\rangle)/\sqrt{2}$$

If applied to $|1\rangle$,

$$(|0\rangle - |1\rangle)/\sqrt{2}$$

Matrix Notation:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

No.
Date

4) Classical Gates

- NOT(X)

Flips the qubit state (like a classical NOT gate).

Effect: $|0\rangle \leftrightarrow |1\rangle$

⇒ Matrix

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- CNOT (Controlled-NOT)

Flips the target qubit if the control qubit is $|1\rangle$.

Effect: works like a classical XOR.

Matrix:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

* Toffoli (CCNOT)

Intuitive way to understand this is it's a AND gate in quantum computing.

Effect: Flips the target qubit only if both control qubits are |1>.

$$Toffoli = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

* SWAP Gate:

Exchanges two qubits.

Effect: $|01\rangle \leftrightarrow |10\rangle$

Matrix:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

No.

Date

Phase Gates

* Z Gate

Leaves $|0\rangle$ unchanged but flips the phase of $|1\rangle$.

Matrix :-

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

* S Gate

A square-root of Z gate, used for phase shifts.

Matrix :-

$$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$$

* T Gate

Intuitively, quarter-turn phase shift.

Matrix :-

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

No

Date

* RZ (Rotation-Z)

Rotates the qubit around the Z-axis by an arbitrary angle θ.

Matrix :-

$$RZ(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix}$$

Quantum Gates :-

→ SX (Square root X)

Halfway between Identity and X.

Matrix :-

$$SX = \frac{1}{2} \begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}$$

→ Y Gate :-

Like X, but also flips the phase.

Matrix :-

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

No.

Date

→ RX (Rotation-X)

Rotates around x-axis by θ.

Matrix :-

$$RX(\theta) = \begin{bmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

→ RY (Rotation Y)

Rotates around Y-axis by θ.

Matrix :-

$$RY(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

→ RXX (Two-qubit X Rotation)

Matrix Notation :-

$$RXX(\theta) = \cos(\theta/2)\, I - i\sin(\theta/2)(X \otimes X)$$

→ RYY (Two-qubit Z Rotation)

Matrix Notation :-

$$RZZ(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 & 0 & 0 \\ 0 & e^{-i\theta/2} & 0 & 0 \\ 0 & 0 & e^{-i\theta/2} & 0 \\ 0 & 0 & 0 & e^{-i\theta/2} \end{bmatrix}$$

No
Date

- U Gate (General Unitary)
  General purpose rotation in Quantum Computing.

Matrix :-

$$U(\theta, \phi, \lambda) = \begin{bmatrix} \cos(\theta/2) & -e^{i\lambda}\sin(\theta/2) \\ e^{i\phi}\sin(\theta/2) & e^{i(\phi+\lambda)}\cos(\theta/2) \end{bmatrix}$$

Additional Gates (Good to know)

Single Qubit

1) Identity Gate (I)
   Keeps the qubit unchanged.

Matrix :-

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

2) T+ (T-dagger) (T dg)
   Inverse of the T gate.

Matrix :-

$$T^+ = \begin{bmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{bmatrix}$$

No

Date

3) S† (S dagger gate)
The inverse of S gate.

matrix :

$$S^\dagger = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}$$

Multi-Qubit Gates :

1) CZ (controlled-Z)
2) CS (controlled-S)
3) CS† (controlled-S†)
4) CCZ, CT, CT†
5) Fredkin gate (controlled-SWAP)
6) Peres (simplified Toffoli gate)

Exotic Gates

1) iSWAP
Swaps two qubits and applies a phase shift.

2) F gate (Fourier Transform Gate)
3) Deutsch Gate (Universal gate that can approximate any quantum computation)
4) Generalized controlled U-gate

No.

Date

How to create your own Gates?

Involves theoretical physics, linear algebra and quantum information theory

S1:- Define the purpose

→ What problem do you want the new gate to solve?

→ Does it simplify an existing operation?

→ Can it speed up a quantum algorithm?

→ Is it useful in a specific quantum computing model (like trapped ions, superconducting qubits, etc.)?

S2:- Define the Mathematical Model:-

⇒ Every quantum gate is represented as a Unitary Matrix U that follows:-

$$U^\dagger U = I$$

No.

Date

→ Use a unitary transformation to construct the gate.

→ Example:- If you want a new kind of rotation, you can modify the standard $R(\theta)$ gates.

S3:- Implement in Code.

S4:- Compute Eigenvalues and Eigenvectors to check how it affects quantum state.

→ Verify its commutative relations with other gates.

→ Simulate how it interacts with and in multi-qubit environments.

S4:- Publish

→ Check if it is unique

→ Find practical use cases.

→ If promising, write a research paper and submit it to quantum computing conferences.

No.

Date

Types of combinations of Quantum Gates :-

(A) Sequential Combination (Cascading Gates)

Gates are applied one after another, just like in classical circuits.

Effect :- The resulting transformation is the product of the matrices of the individual gates.

Example :-

Applying a Hadamard (H) followed by an X gate.

$$H \cdot X = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Creates a superposition and then flips the states.

(B) Parallel Combination (Tensor Product)

Two gates act on separate qubits simultaneously.

Effect:- The operation is described using the kronecker (tensor) product.

Example:-

A Hadamard on the first qubit and an identity gate on the second qubit:-

$$H \otimes I = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The first qubit undergoes superposition while the second remains unchanged.

(c) Conditional Combination (Controlled Gates)

A gate is applied to one qubit depending on the state of another.

Effect:- Used in entanglement, conditional logic and quantum circuits.

Example:- CNOT Gate.

No.

Date

Common functionalities of Gate Combinations

uses :- (Perform)

1) Computational
2) Logical
3) Transformation functions

(A) Superposition creation

→ Gate used : Hadamard (H)

→ Places qubits into a equal superposition of $|0\rangle$ and $|1\rangle$.

Mathematically,

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

Now, Qubit is in equal probability of being 0 or 1.

(B) Entanglement Generation

→ Gates used :- H + CNOT

→ Creates entanglement, a key quantum effect.

No.

Date

Mathematically :-

$$|\Psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Apply H on qubit 1 and CNOT on (qubit 1 → qubit 2)

The two qubits are now inseparable - measuring one determines the other.

(c) Quantum Teleportation :-

Gates used :- H, CNOT, Measurement, Classical communication.

→ Transfers a qubit's state to another qubit without direct interaction.

→ Example :-

Uses Bell states and classical correction gates (X, Z) to reconstruct the qubit state at a different location.

No.

Date

(5) Quantum fourier transform (QFT)

Gates used :- H, controlled phase (CRk)

-) performs efficient fourier Transformation a key operation in shor's Algorithm.

Mathematically,

Circuit applies Hadamard and rotation gate in sequence.

$$QFT|x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i x k / N} |k\rangle$$

Summary Table :-

Combination	Functionality	Gates used
1) Grover's Algorithm	Quantum Search Speedup	H, Oracle, Diffusion.
2) Quantum Error correction	Protects qubit from decoherence	CNOT, H, Measurement
3) Quantum Cryptography	Secure communication (BB84)	H, Measurement
4) Quantum Chemistry (VQE)	Simulates molecular properties.	parametric U, Controlled Gates.

No.

Date

Quantum System Encoding Procedure.

1. Convert the problem into mathematical form (Boolean logic, Matrices, Differential Equations).

2. Encode the problem into quantum states (Binary representation, Amplitude encoding).

3. Apply Quantum gates to process and manipulate Qubits.

4. Use measurement to extract the correct solution with high probability.

No.

Date

Encoding a problem into a Quantum Circuit

For SAT (Boolean Satisfiability Problem)

S1 :- Define the problem mathematically.

"Is there a code to assign TRUE / FALSE values to variables to satisfy a logical expression?"

(1) Convert the problems to Qubits

- We say (let) that we have 3 boolean variables (x_1, x_2, x_3)

- We assign one qubit per variable.

(2) Set up the Quantum Circuit

- Hadamard Gate :- Create superposition of all possible solutions.

- Controlled Phase Gates :- Encode logical constraints.

- Quantum Grover Algorithm :- Amplify the correct solution.

(3) Measure the Qubits

- The correct solution appears with high probability.

No.

Date

Quantum Teleportation

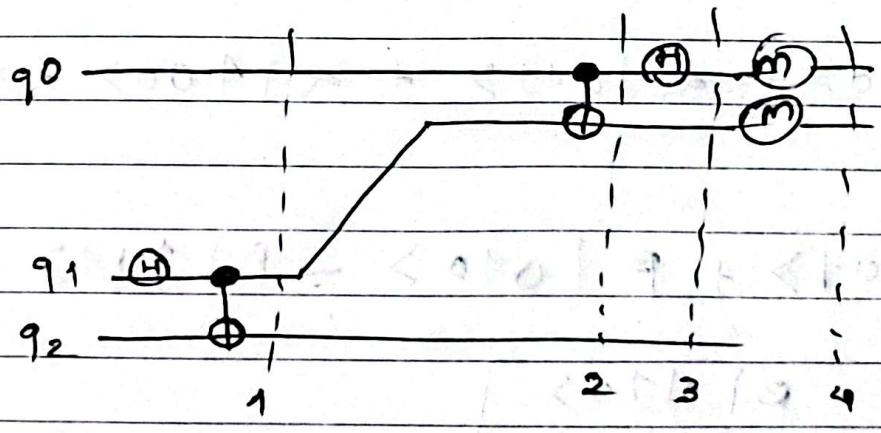
* Z-gate (phase flip)

$|0\rangle \rightarrow |0\rangle$

$|1\rangle \rightarrow -|1\rangle$

* Quantum Teleportation:-

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ Alice



$\frac{1}{\sqrt{2}} |00\rangle + |11\rangle$ Bob

Total Quantum state

phase 1

$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \otimes \frac{1}{\sqrt{2}} |00\rangle + |11\rangle$

$= \frac{1}{\sqrt{2}} [\alpha|000\rangle + \alpha|011\rangle +$

$\beta|100\rangle + \beta|111\rangle]$

No.

Date

Phase 2 :

$$\frac{1}{\sqrt{2}} [\alpha|000\rangle + \alpha|011\rangle + \beta|101\rangle + \beta|110\rangle]$$

Phase 3 :

$$\frac{1}{\sqrt{2}} [\alpha|000\rangle + \alpha|011\rangle + \alpha|100\rangle + \alpha|111\rangle + \beta|001\rangle + \beta|010\rangle - \beta|101\rangle - \beta|110\rangle]$$

Phase 4 (Measurement) :

q_0, q_1	00	01	10	11
q_2 :	$\alpha\|0\rangle + \beta\|1\rangle$	$\alpha\|1\rangle + \beta\|0\rangle$		
	$\alpha\|0\rangle - \beta\|1\rangle$	$\alpha\|1\rangle - \beta\|0\rangle$		

No.

Date

⇒ Superdense coding :

Superdense coding is a quantum communication protocol that enables the transmission of two classical bits of information using only one qubit, provided that the sender and receiver initially shared an entangled pair of qubits (a Bell state).

⇒ Quantum Teleportation :

Quantum Teleportation is a protocol for transferring an unknown quantum state from one location to another, using two classical bits and a pre-shared entangled pair of qubits. The original state is destroyed during the process, respecting the no-cloning theorem.

The teleportation protocol can be summarized mathematically :

$$|\psi\rangle \otimes |B_{00}\rangle = \frac{1}{2}\Big[|B_{00}\rangle(\alpha|0\rangle + \beta|1\rangle) + |B_{01}\rangle(X|\psi\rangle) + |B_{10}\rangle(Z|\psi\rangle) + |B_{11}\rangle(XZ|\psi\rangle)\Big]$$

No.

Date

Bernstein - Vazirani Algorithm:-

Problem statement:-

Given a Boolean function $f:\{0,1\}^n \to \{0,1\}$, promised to be of the form:-

$$f(x) = q \cdot x = (q_1 x_1 \oplus q_2 x_2 \oplus \ldots \oplus q_n x_n) \mod 2$$

where,

$q = (q_1, q_2 \ldots q_n)$ is an unknown secret bit-string we want to find.

$x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ is the input string.

$\oplus$ represents additional modulo 2 (XOR operation)

The goal of the Bernstein-Vazirani algorithm is to determine the secret bit string q with only one query to the oracle function $f(x)$.

Mathematical formulation of Bernstein-Vazirani Algorithm:-

Step 1:- Initialization

Start with an n+1 qubit state initialized as :-

$$|0\rangle^{\otimes n} |1\rangle$$

S2:- Apply Hadamard Gates:-

Apply Hadamard gates $H^{\otimes n}$ to the first register and a Hadamard gate to the second register as :-

→ First register transform from $|0\rangle^{\otimes n}$ to :-

$$H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n - 1} |x\rangle$$

→ Second Register transform from $|0\rangle$ to :-

$$H|0\rangle = |+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$$

Thus, the initial state after Hadamard gate is :-

$$|\psi_{init}\rangle = \frac{1}{\sqrt{2^n}} \sum_{x}^{2^n - 1} |x\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

3x+1 3x+1

No.

Date

5 5 :- Oracle operation

The oracle U_f acts as computational basis states as follows :-

$$|x\rangle|y\rangle \xrightarrow{U_f} |x\rangle|y \oplus f(x)\rangle$$

When applied to our state, this oracle induces a phase kickback into the first register.

Specifically, when second register is in state :-

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = H|1\rangle,$$

applying oracle yields :-

$$U_f(|x\rangle|-\rangle) = (-1)^{f(x)}|x\rangle|-\rangle$$

Thus our state after oracle becomes :-

$$|\Psi_{oracle}\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{f(x)}|x\rangle$$

$$= \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} (-1)^{a \cdot x}|x\rangle$$

No.

Date

Now, applying this transformation gives us:-

$$|\psi_{final}\rangle = \frac{1}{2^n} \sum_{x=0}^{2^n-1} \sum_{y=0}^{2^n-1} (-1)^{a \cdot x + x \cdot y} |y\rangle$$

$$= \sum_{y=0}^{2^n-1} \left(\frac{1}{2^n} \sum_{x=0}^{2^n-1} (-1)^{(a+y) \cdot x} \right) |y\rangle$$

The inner summation can be simplified using the identity:-

for binary strings a, y, x:-

If $a = y$, then for all bits:-

$$x_i y_i = 0 \pmod 2,$$

thus, sum over all possible x:

equals 2^n.

If $a \neq y$, then half the terms are positive and half negative, summing exactly to zero.

Hence we have,

$$\forall\, y \neq a : \text{amplitude} = 0$$
$$\forall\, y = a : (a+a) = 0 \pmod 2,$$

No

Date

the amplitude is 1.

Therefore

$$|\psi_{final}\rangle = |a\rangle.$$

Bell States

* Bell state is a maximally entangled quantum state of 2 qubits. The four canonical Bell states are:

$$|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

$$|\phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$$

$$|\psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$$

$$|\psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$$

No.

Date

Deutsch's Algorithm

	f(0)	f(1)	
1	0	0	Constant
2	0	1	Balanced
3	1	0	Balanced
4	1	1	Constant

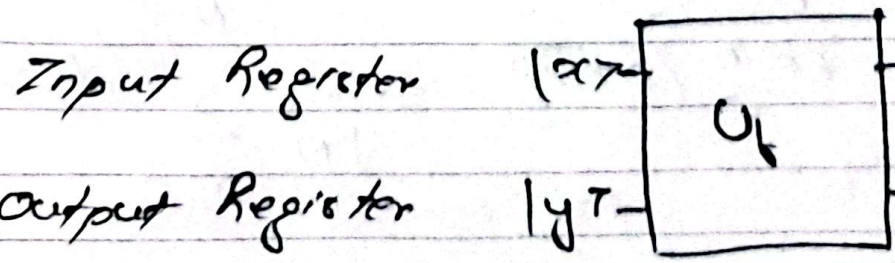
Find if the function is constant or balanced.

$$f : \{0,1\} \rightarrow \{0,1\}$$

$$f(0) = f(1) = \text{constant}$$

Deutsch Jouza is for n^{th} classification.

→ Circuit design



$$|x\rangle|y\rangle \xrightarrow{U_f} |x\rangle|y \oplus f(x)\rangle \xrightarrow{H} |x\rangle|y \oplus f(x)\rangle$$

$$\oplus f(x)\rangle$$

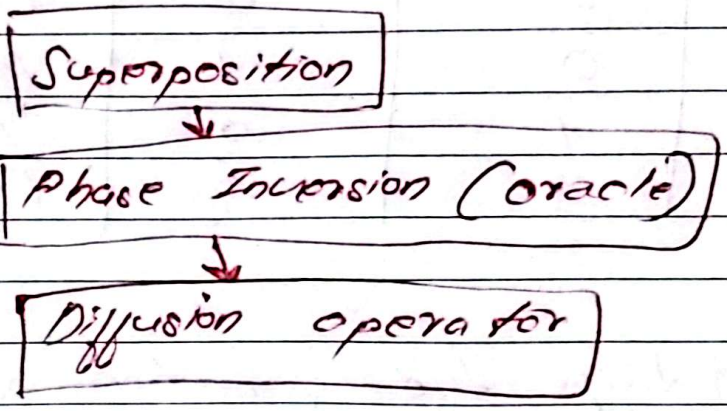
$$|x\rangle|y \oplus 0\rangle \; |x\rangle|y\rangle.$$

No.

Date

Grover's Algorithm

Intuition :-



* Mathematical formulation:-

problem statement :- Given an unsorted database of $N = 2^n$ elements, find one marked element as efficiently.

Represent the database as basis states :-

$$|0\rangle, |1\rangle, |2\rangle, \ldots, |N-1\rangle$$

Initially, create equal superposition over all states using Hadamard gates :-

$$|\psi\rangle = H^{\otimes n} |0\rangle^{\otimes n} = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

No.

Date

⇒ Grover's Iteration (core step)

Each iteration has two main operations :-

① Oracle Operator (U_ω) :-

Flips the phase of the target state $|x_0\rangle$:-

$$U_\omega |x\rangle = \begin{cases} -|x\rangle & x = x_0 \\ |x\rangle & x \neq x_0 \end{cases}$$

Mathematically represented as :-

$$U_\omega = I - 2|x_0\rangle\langle x_0|$$

② Diffusion Operator (Amplitude Amplification)

Reflects about the average amplitude :-

$$U_s = 2|\psi\rangle\langle\psi| - I$$

where,

$$|\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

Combined Grover Iteration operator :-

$$G = (2|\psi\rangle\langle\psi| - I)(I - 2|x_0\rangle\langle x_0|)$$

Complexity :- $\sqrt{N}$

No.

Date

Shor's Algorithm:-

* Problem Statement:-

Given an integer N, find it's prime factor efficiently.

Classically, factoring large integer is computationally expensive. Shor's algorithm solves factoring efficiently on Quantum Computers.

* Note:- This is classical pre and post processing.

Mathematical Formulation:-

To factor integer N:-

(1) Choose a random integer a, where

$1 < a < N$, and compute greatest common divisor (GCD)

→ If $gcd(a, N) \neq 1$, you've found a factor easily.

→ Otherwise, proceed to step 2.

(2) Define the function:-

$$f(x) = a^x \bmod N$$

No.

Date

If we can find this period r, we can factor N easily using classical methods like (Euclidean algorithm).

→ If r is even, and $a^{r/2} \neq -1 \bmod N$, then we can compute factors by :-

$$\gcd(a^{r/2} - 1, N) \quad \text{and}$$

$$\gcd(a^{r/2} + 1, N)$$

Post-Classical processing re-visited later too

(#) To find period r, we use Quantum Algorithm :-

* Quantum Order finding subroutine (Quantum part)

This step uses Quantum phase Estimation :-

⇒ Quantum Registers Setup :-

We use two quantum registers :-

→ First register :- consists of $2n$ qubits initialized in state :-

$$|0\rangle^{\otimes 2n}$$

where, $n = \lceil \log_2(N) \rceil$

No.

Date

→ Second register :- consists of n qubits initialized in state :-

$$|1\rangle$$

Thus, the initial state is :-

$$|\psi_0\rangle = |0\rangle^{\otimes 2n} |1\rangle$$

↓ Step by step Quantum Procedure :-

S1 :- Apply Hadamard Gates :-

Apply Hadamard gates to the first register creating equal superposition.

$$|\psi_1\rangle = \frac{1}{\sqrt{2^{2n}}} \sum_{x=0}^{2^{2n}-1} |x\rangle |1\rangle$$

S2 :- Oracle Operation (Modular Exponentiation)

Apply the oracle operator U_f, designed to perform modular exponentiation :-

$$U_f |x\rangle |y\rangle = |x\rangle |y + f(x) \bmod N\rangle$$

No.

Date

After applying oracle to second register, it is initialized as $|1\rangle$:-

$$|\psi_2\rangle = \frac{1}{\sqrt{2^{2n}}} \sum_{x=0}^{2^{2n}-1} |x\rangle |a^x \pmod N\rangle$$

S3 :- Quantum Fourier Transform (QFT)

Apply Quantum Fourier Transform to first register. The QFT transforms computational basis states as follows :-

$$|x\rangle \xrightarrow{QFT} \frac{1}{\sqrt{2^{2n}}} \sum_{y=0}^{2^{2n}-1} e^{2\pi i xy/2^{2n}} |y\rangle$$

Applying QFT after oracle gives us :-

Interference patterns encoding periodicity information.

S4 :- Measurement and Period Extraction.

Measures the first register. The measurement yields an integer value close to multiples of :-

No.

Date

$$y = \left[\frac{j}{r} 2^{2n} \right], \quad j = 0, 1, \ldots, r-1$$

Thus, the measurement output encodes information about the period r.

* Classical post-processing (continuous fraction)

The measured value of the quantum circuit is an integer approximation to:

$$\frac{j}{r} 2^{2n}$$

Use classical algorithm called "continuous fractions" to extract exact values of numerator and denominator $\left(\frac{j}{r}\right)$ from this approximation. Specifically,

Given integer (measured) k, compute fraction:

$$\frac{k}{2^{2n}} \approx \frac{j}{r}$$

"Using continued fraction algorithm, you can precisely recover integers j, r. once you

have period r:

If r is even and $a^{r/2} \neq -1 \pmod{N}$, compute factors by classical GCD method:

Factors are given by:

→ Factor #1: $p = \gcd(a^{r/2} - 1, N)$

→ Factor #2: Compute similarly with $a^{r/2} + 1, N$

If conditions fails (odd period or trivial factors), repeating with different random choice of a.

Summary

Step	Operation	Mathematically
1) classical step	Pick random number & check GCD	Choose random $a < N$, $GCD(a, N) \neq 1$?
2) Quantum step	Create superposition	Apply Hadamard Gate on first register.
3) Quantum oracle	Modular exponential oracle	Compute $x^a \bmod N$ in second register
4) Quantum Fourier Transform	Apply QFT on first register.	Extract periodicity information.

No.

Date

5) Measurement and Classical processing.	Measure and continued fraction algorithm.	Recover period r: $a^r = 1 \pmod N$
6) Factorization	Compute GCDs classically	Factors from: $\gcd(a^{r/2} \pm 1, N)$

Shor Algorithm In-depth

→ Algorithm:

1) Modular Arithmetic
2) Divisor (GCD)
3) Periodicity

→ General Rule:

$$x! = 1 - 1 \pmod N$$
$$x^2 = 1 \pmod N$$
$$\gcd(N, x+1)$$

No.

Date

Quantum Fourier Transform (QFT)

Modular Exponentiation -

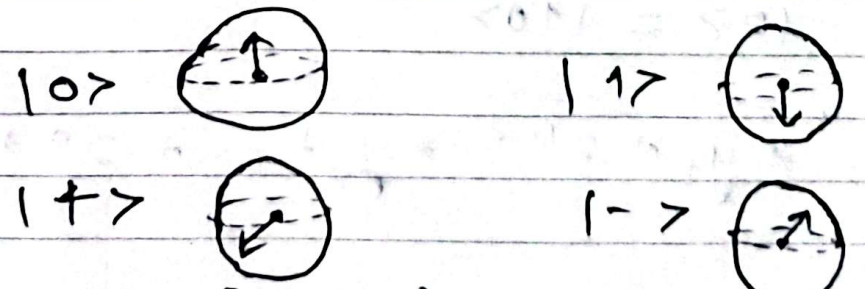
Quantum Phase Estimation (QPE)

→ Quantum Fourier Transform :-

1 (Qubit)

Computational Basis —H→ Fourier Basis

$\{|0\rangle, |1\rangle\}$ $\{|+\rangle, |-\rangle\}$



$1/\sqrt{2}(|0\rangle + |1\rangle)$ $1/\sqrt{2}(|0\rangle - |1\rangle)$

→ Quantum Fourier Transform :-

(2 Qubit)

Computational Basis —→ Fourier Basis

$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$

No

Date

$$QFT|x\rangle = |\tilde{x}\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{\frac{2*pi*i*x*y}{N}} |y\rangle \quad \text{---(i)}$$

Note: $e^{pi\,x\,i} = -1$

$N = 2^n$

n = # qubits

where,

$y = [y_1, y_2, y_3, \cdots y_n]$

$|0\rangle, |1\rangle, |2\rangle, |3\rangle$

$|2\rangle = |10\rangle$

$= y^{n-1} * y_1 + 2^{n-2} * y_2 + \cdots + 2^0 * y_n$

So,

$$y = \sum_{k=0}^{n} y_k 2^{n-k} \quad \text{---(ii)}$$

Substituting value of y in (ii) to (i)

$$QFT|x\rangle = |\tilde{x}\rangle = \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} e^{2\pi i x \sum_{k=0}^{n} y_k/2^k}$$

$|y_1, y_2, y_3, \cdots, y_n\rangle$

No

Date

$$= \frac{1}{\sqrt{N}} \sum_{y=0}^{N-1} \prod_{k=1}^{n} e^{\frac{2\pi i x y_k}{2^k}} |y_1, y_2, \cdots y_n\rangle$$

Therefore

$$= \frac{1}{\sqrt{N}} \left(|0\rangle + e^{\frac{2*\pi i * 1 x}{2^1}} |1\rangle\right) \otimes$$

$$\left(|0\rangle + e^{\frac{2\pi i x}{2^2}} |1\rangle\right) \otimes \quad \cdots$$

$$\left(|0\rangle + e^{\frac{2\pi i x}{2^n}} |1\rangle\right)$$

Into matrix form:-

$$\begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{2\pi i}{2^k}} \end{bmatrix}$$

This is a quantum gate (should be comparable to U General gate)

No.

Date

Quantum Phase Estimation.

$U|\psi\rangle = e^{i\theta}|\psi\rangle$

↳ Finding this -