Order Finding

Find a modular multiplication period by estimating eigenphases and reconstructing a verified rational denominator.

Undergraduate 22 min lesson Dr. Charith Jayasekara

By the end, you'll be able to

- Calculate a modular orbit and its order.
- Explain why sampled reduced denominators may divide the true order.
- Use repeated squaring and verification in the order-finding workflow.

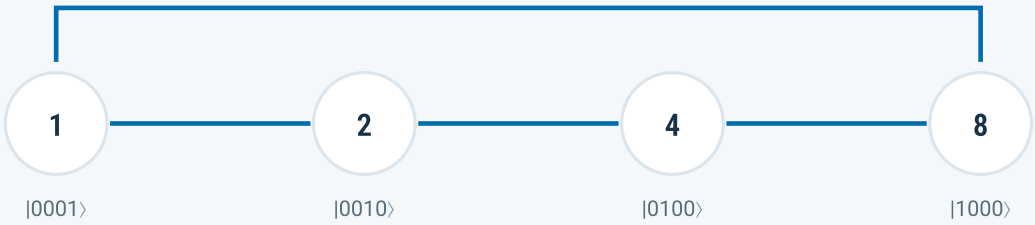
Find when modular multiplication repeats

Order finding asks how many times a number must be multiplied by itself before its remainder returns to one. For integers a and N with no common factor, the order r is the smallest positive integer satisfying a to the power r congruent to one modulo N. Quantum order finding turns this repetition into an eigenphase-estimation problem.

Review phase estimation and the QFT. The classical prerequisites are modular arithmetic, greatest common divisors and fractions. You do not need advanced number theory to follow the small example; the important step is separating the quantum phase sample from the classical recovery and verification of r.

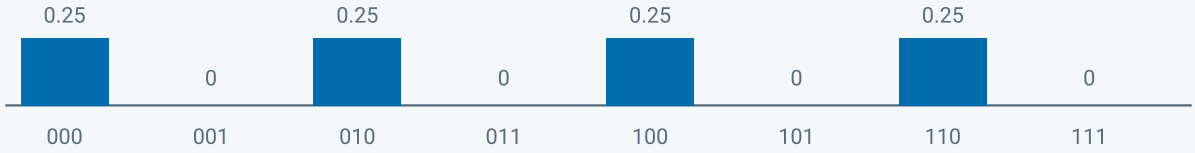
WORKED DIAGRAM · A13

Order finding: a cycle becomes four eigenphases



Multiply by 2 modulo 15 · after 4 steps return to 1

Fourier peaks



Probability · scale 0 to 1

Multiplication by two modulo fifteen cycles 1→2→4→8→1. Input |1⟩ overlaps four orbit eigenstates equally. QPE samples phases 0, 1/4, 1/2, 3/4, and candidate orders require modular verification.

What goes in, what comes out, what we assume

The inputs a and N are ordinary classical integers with greatest common divisor one. The desired output is the smallest positive order r. The quantum procedure uses reversible modular multiplication on a register holding remainders. The relevant unitary maps y to a times y modulo N on valid residue labels, with a reversible extension on any unused computational-basis labels.

Because a is coprime to N, multiplication permutes the residues and is invertible. If the common divisor were greater than one, that permutation construction would fail, but in factoring the divisor would already be useful classical information. Check the coprimality condition before applying the quantum subroutine.

The access model is more concrete than an arbitrary truth-table oracle: modular powers can be computed by structured arithmetic. A full algorithm must implement that arithmetic reversibly, account for workspace and uncompute temporary information. The quantum register width alone does not determine its gate cost.

The order is the first positive exponent that returns one. Coprimality makes multiplication a reversible permutation.

$$r = \min\{k \geq 1 : a^k \equiv 1 \pmod{N}\}, \qquad U_a|y\rangle = |ay \bmod N\rangle \quad (0 \leq y < N).$$

Assume $\gcd(a, N) = 1$ and extend U_a reversibly to any unused binary labels. The input numbers a, N are classical; the register holds residues.

Worked example: powers of two modulo fifteen

The sequence begins one, two, four, eight, then one again. The order is four. Define U to multiply by two modulo fifteen. On the orbit containing one, U cycles through four basis states. Linear combinations of those orbit states form eigenstates with eigenvalues corresponding to the four quarter-turn phases.

The easy-to-prepare basis state one is an equal superposition of those four eigenstates. Phase estimation therefore samples phases zero, one quarter, one half and three quarters with equal probability in an exactly resolved ideal example. If the counting grid has eight points, the numeric labels are zero, two, four and six.

The one-quarter sample has reduced denominator four, suggesting r equal to four. The three-quarter sample does too. The one-half sample has denominator two, which is only a divisor of the true order. Checking two squared modulo fifteen gives four, not one, so two is not an acceptable final answer. The zero sample gives no useful denominator for this run. Repetition and classical reconstruction are essential parts of the procedure.

The orbit of one is a four-state cycle, which supplies four equally weighted phase branches.

$$1 \xrightarrow{\times 2 \bmod 15} 2 \longrightarrow 4 \longrightarrow 8 \longrightarrow 1, \qquad r = 4, \qquad \phi \in \left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}\right\}.$$

With a three-bit counting register, those phases appear at numeric labels $0, 2, 4, 6$, each with ideal probability $1/4$.

Checkpoint 1

PAUSE & THINK

A run suggests phase 1/2. Does denominator two establish the order of 2 modulo 15?

- [ ] Yes, because every denominator is the order
- [ ] No; 2^2 mod 15 is 4
- [ ] No; orders must always be odd

Answer: No; 2^2 mod 15 is 4. The true order is four. Phase 2/4 reduces to 1/2, so this branch alone loses a factor of two in the denominator.

Why the phases encode the order

On a cycle of length r, the modular multiplication operator is a cyclic shift. Its eigenvectors are Fourier combinations around that cycle, and their eigenvalues are exp of two pi i j over r for j from zero to r minus one. Preparing the orbit's first basis state gives overlap with all those eigenvectors.

QPE samples an approximation to j over r. If j and r are coprime and the approximation is sufficiently precise, continued fractions can recover the reduced denominator r. If they share a factor, the reduced denominator can be smaller. Combining information from runs and verifying candidate orders addresses that issue. The algorithm learns a period through a spectrum, not by waiting for a measured trajectory to complete r physical steps.

Fourier combinations around the orbit are eigenstates of modular multiplication.

$$|u_j\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} e^{-2\pi i j k/r} |a^k \bmod N\rangle, \quad U_a|u_j\rangle = e^{2\pi i j/r}|u_j\rangle, \quad |1\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} |u_j\rangle.$$

$j = 0, \ldots, r-1$ labels the eigenphase branches. The negative sign in the eigenvector definition gives the positive eigenvalue convention used by QPE.

Recover a rational number carefully

Continued fractions express a real approximation through simple rational convergents. In order finding, the denominator is known to be bounded by N, and sufficiently accurate phase estimation makes the relevant rational distinguishable. A familiar sufficient closeness scale is less than one over twice the square of the true denominator. This explains why the counting precision grows with the bit-length of N.

The measured grid value is not always exactly j over r. For an order such as six, fractions like one sixth cannot be represented exactly by a power-of-two grid. The output distribution concentrates near them, and the classical rational reconstruction uses that approximation. A tiny textbook instance whose period divides the grid hides this important finite-resolution step.

Do not simply take the denominator of the raw binary-grid fraction. That denominator is usually a power of two chosen by the experiment, not the unknown order. Continued fractions search for a nearby small-denominator rational, and modular exponentiation verifies whether a proposed period actually returns one.

A sufficiently close phase estimate identifies a small-denominator rational. Its denominator still needs a modular check.

$$\left|\widehat{\phi} - \frac{j}{r}\right| < \frac{1}{2r^2}, \qquad \frac{j}{r} = \frac{p}{q} \text{ in lowest terms}, \qquad q = \frac{r}{\gcd(j, r)}.$$

For a reduced rational, the closeness bound ensures it occurs among the continued-fraction convergents. If j and r share a factor, q may be smaller than the true order: $\phi = 1/2$ in this example suggests $q = 2$, but $2^2 \bmod 15 = 4 \neq 1$.

Implement controlled powers efficiently

Canonical QPE calls for U raised to powers of two. Repeating a base multiplication that many times would suggest an exponential circuit. Instead, compute constants a raised to those powers modulo N classically by repeated squaring, then build controlled modular multiplication by each constant. The arithmetic has structure that permits polynomial-size reversible implementations in the input bit-length.

This is a crucial example of an algorithm-specific power shortcut. General phase estimation does not guarantee cheap access to every high power of an arbitrary unitary. Order finding supplies a mathematical representation that makes the required powers efficient. Its advantage depends on that structure and on reliable execution of the arithmetic, not only on a Fourier transform.

Repeated squaring supplies each controlled multiplier directly.

$$b_0 = a \bmod N, \qquad b_{\ell+1} = b_\ell^2 \bmod N, \qquad U_a^{2^\ell}|y\rangle = |b_\ell y \bmod N\rangle.$$

For $a = 2, N = 15$, the first constants are $2, 4, 1, 1, \ldots$. Compute these constants from the input and implement reversible multiplication; the circuit does not need the unknown order inserted into it.

Checkpoint 2

PAUSE & THINK

Why is a high modular power efficient here?

- [ ] Its constant multiplier can be computed by repeated squaring
- [ ] Every arbitrary unitary has cheap powers
- [ ] The circuit assumes r is known

Answer: Its constant multiplier can be computed by repeated squaring. Structured modular arithmetic supplies the shortcut. This is an algorithm-specific advantage, rather than a generic property of QPE.

Complexity and interpretation

The input size is the number of bits needed for N, not the magnitude of N. Quantum order finding has polynomial algorithmic cost in that bit-length using appropriate arithmetic and sufficient precision. Physical resource estimates add error correction, gate synthesis, timing and success amplification. Those additions are major engineering questions for cryptographically sized numbers.

This subroutine returns an order, not automatically a factor. Shor's factoring reduction uses a suitable even order and greatest-common-divisor calculations to obtain factors. Some choices of a are unhelpful, so the larger factoring algorithm includes classical preprocessing and repetition. Keeping the subroutine task description narrow makes the overall proof easier to follow.

Explore the mechanism

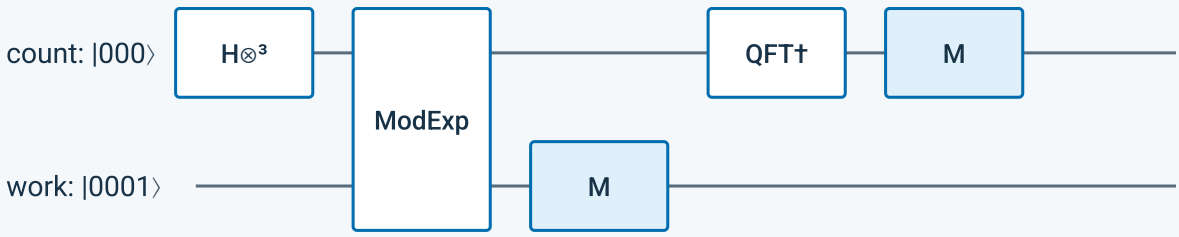
Use Explore freely to compare amplitudes and measurement probabilities. Choose Predict & test for a quick prediction, then run measurements. Change one setting at a time and explain the change.

ALGORITHM LAB

Ideal local model

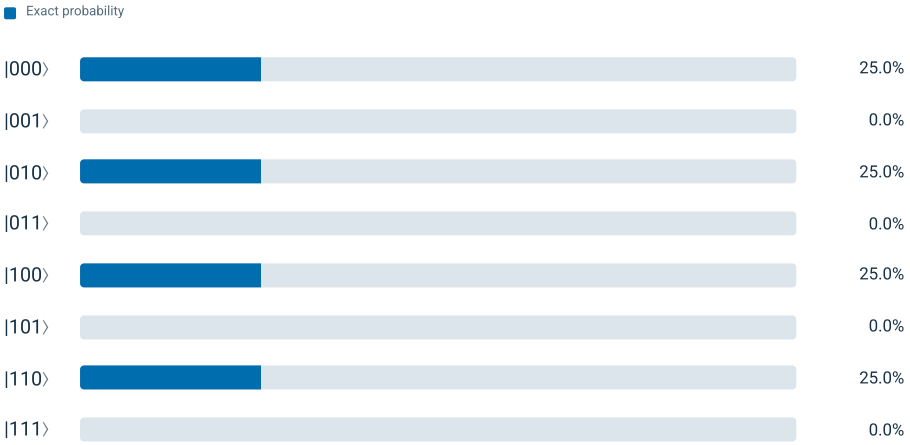
Order finding: modular powers and interference

The upper drawn wire is a three-qubit counting register and the lower is a four-qubit work register. Modular powers are computed for N=15; the output shown is conditioned on the selected work result.



This small modular-power circuit derives its amplitudes from computed values; larger factoring tasks need more precision and scalable reversible arithmetic.

Initial model reference



Statevector before readout · ideal simulation

Basis	Amplitude a + bi	Magnitude	Phase (rad)	Probability
|000⟩	`0.5 + 0i`	0.500000	0.000000	25.0%
|001⟩	`0 + 0i`	0.000000	undefined	0.0%
|010⟩	`0.5 + 0i`	0.500000	0.000000	25.0%
|011⟩	`0 + 0i`	0.000000	undefined	0.0%
|100⟩	`0.5 + 0i`	0.500000	0.000000	25.0%
|101⟩	`0 + 0i`	0.000000	undefined	0.0%
|110⟩	`0.5 + 0i`	0.500000	0.000000	25.0%
|111⟩	`0 + 0i`	0.000000	undefined	0.0%

Signs and complex phases are part of the state. Phase is undefined for a zero amplitude. A shared global phase changes no probabilities; relative phases can affect a later operation. Values are rounded to six decimal places.

Chapter summary

- Order finding learns the period of reversible modular multiplication.

- An easy basis input samples Fourier eigenphases j/r.
- Use rational reconstruction and modular verification; one reduced denominator need not be the full order.

A common misconception

The denominator of any measured phase is automatically the order. The raw binary-grid denominator belongs to the chosen precision. A nearby reduced rational can have denominator dividing the true order; it needs reconstruction and verification.

Code practice

Download the tested Qiskit notebook and follow the pinned environment instructions. Inspect the circuit, predict a result and run its independent assertions. The modular-action example is a small-instance demonstration; the notebook states its precision and post-processing limits.

Exercises

1. Compute the order of two modulo seven by listing successive powers. Does the order have to be even?
2. In the fifteen example, QPE returns a phase of one half. Why is reporting order two immediately incorrect, and what verification detects the problem?

> Worked answers
>
> 1. The powers are two, four and one modulo seven, so the order is three. Orders need not be even. Factoring reductions therefore have conditions beyond merely obtaining some order.
> 2. The reduced denominator can be a divisor of the true order when the sampled numerator shares a factor with it. Two squared modulo fifteen is four, so the candidate fails the defining order condition. Another run yielding one quarter or three quarters reveals denominator four, which passes verification.

Check your understanding

PAUSE & THINK

QPE returns phase 1/2 for the order of two modulo fifteen. Why must denominator two be checked?

- [ ] It may be only a divisor of the true order
- [ ] Every measured phase is exact zero
- [ ] An order cannot be even

Answer: It may be only a divisor of the true order. The sampled numerator and true order can share a factor. Two squared is four modulo fifteen, so order two fails verification; the true order is four.

Sources

Shor's original paper develops the polynomial-time arithmetic and factoring framework. IBM's Shor lesson explains order finding through phase estimation and classical reconstruction.