ALGORITHM :

Integer Factorization

Input: A positive integer $N \geq 2$

Output: A prime factorization $N = p_1^{k_1} \times \cdots \times p_m^{k_m}$.

Method:

POLYNOMIAL TIME →

```
if (N = p^k , where p is prime , k ≥ 1)
    return p^k
else
```

POLYNOMIAL TIME →

```
    if (N is even) {
        return combine (2 , IntFac (N/2))
    }
    else {
        int d = Shor (N)      ← a factor of N
        return combine (IntFac(d), IntFac(N/d))
    }
```

Shor's Algorithm:

Input: An odd, composite integer N; not power of a prime.

Output: A nontrivial factor, d of N.

$$1 < d < N, \quad d \mid N$$

should divide

Method:

repeat

  int a = random $\{2, \dots, N-1\}$

  int d = $\gcd(a, N)$

  if $(d > 1)$ {return d}

  else {

   int r = FindOrder(a, N) Smallest r such that $a^r \equiv 1 \pmod N$

   if (r is even) {

    int $x = (a^{r/2} - 1) \pmod N$

    int $d = \gcd(x, N)$

    if $(d > 1)$ return d

   }

  }

until we give up

$\rightarrow \mathbb{Z}_N = \{0, 1, \dots, N-1\}$

$\mathbb{Z}_N^* = \{a \in \mathbb{Z}_N \mid \gcd(a, N) = 1\}$

↑ group

$\rightarrow$ if $a^r \equiv 1 \pmod N$

$N \mid (a^r - 1)$

↑ as r is even

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

$\rightarrow$ So does N divide $a^{r/2} - 1$

if N did divide $a^{r/2} - 1$

$a^{r/2} \equiv 1 \pmod N$

then $r/2$ is the smallest r such that $a^r \equiv 1 \pmod N$

CONTRADICTION

So N does not divide $a^{r/2} - 1$

One iteration success:

$$\boxed{P(\text{success}) \geq 1/2}$$

FIND ORDER

$\hookrightarrow$ CLASSICAL POLYNOMIAL IN N

$\hookrightarrow$ QUANTUM POLYNOMIAL IN $\log N$ //

EXAMPLE:

$N = 21$

Shor (21)

1. pick $a = 2$

$d = \gcd(a, N) = \gcd(2, 21) = 1$

2. FindOrder (a, N)

$= \text{FindOrder}(2, 21)$

$2^6 = 64$

$64 \equiv 1 \pmod{21}$

$r = 6$

3. $a^{r/2} - 1 \pmod{N}$

$= 2^3 - 1 \pmod{21}$

$= 7$, which is a factor of 21.

21 divides $2^6 - 1 = 63$

$$(2^3+1)(2^3-1)$$

$$9 \qquad 7$$

ORDER FINDING ALGORITHM:

INPUT An integer N, an element $a \in \mathbb{Z}_N^*$

OUTPUT The smallest $r > 0 : a^r \equiv 1 \pmod N$

Define unitary

$$M_a |x\rangle = |ax\rangle$$

binary

$ax \pmod N$

↳ understood from now.

$$|\psi_k\rangle = \frac{1}{\sqrt{r}} \left(|1\rangle + \omega^{-k} |a\rangle + \omega^{-2k} |a^2\rangle + \dots + \omega^{-(r-1)k} |a^{r-1}\rangle \right)$$

Eigenvectors of M_a

where $\omega = e^{2\pi i / r}$

$k=1$

Show $|\psi_1\rangle$ is an eigenvector of M_a.

$$M_a|\psi_1\rangle = M_a \frac{1}{\sqrt{r}}\Big(|1\rangle + \omega^{-1}|a\rangle + \omega^{-2}|a^2\rangle + \dots +$$
$$\omega^{-(r-1)}|a^{r-1}\rangle\Big)$$

$$= \frac{1}{\sqrt{r}}\Big(|a\rangle + \omega^{-1}|a^2\rangle + \omega^{-2}|a^3\rangle + \dots$$
$$+ \omega^{-(r-1)}|a^r\rangle\Big)$$
$$\hookrightarrow 1$$

We want to find r such that $a^r \equiv 1 \pmod N$

$$= \frac{1}{\sqrt{r}}\Big(|a\rangle + \omega^{-1}|a^2\rangle + \omega^{-2}|a^3\rangle + \dots$$
$$+ \omega^{-(r-1)}|1\rangle\Big)$$

$$= \frac{\omega}{\sqrt{r}}\Big(\omega^{-1}|a\rangle + \omega^{-2}|a^2\rangle + \omega^{-3}|a^3\rangle + \dots$$
$$+ \omega^{-r}|1\rangle\Big)$$

$$\omega = e^{2\pi i/r}$$

$$\omega^{-r} = \left(e^{2\pi i/r}\right)^{-r} = e^{-2\pi i} = \cos(-2\pi)$$
$$+ i\sin(-2\pi)$$

$= 1 //$

$$M_a |\psi_1\rangle = \frac{\omega}{\sqrt{r}} \left(|1\rangle + \omega^{-1} |a\rangle + \omega^{-2} |a^2\rangle + \omega^{-3} |a^3\rangle + \dots + \omega^{-(r-1)} |a^{r-1}\rangle \right)$$

$$M_a |\psi_1\rangle = \underbrace{\omega}_{\text{eigenvalue}} \overbrace{|\psi_1\rangle}^{\text{eigenvector}}$$

MORE GENERALLY:

$$M_a |\psi_k\rangle = \omega^k |\psi_k\rangle$$

PHASE ESTIMATION ALGORITHM:

INPUT: A unitary U, eigenvector $|\psi\rangle$ of U,

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

OUTPUT: θ

So if we use M_a as U

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

$$M_a |\psi_k\rangle = \omega^k |\psi_k\rangle$$
$$= (e^{2\pi i / r})^k |\psi_k\rangle$$

$$= \left(e^{2\pi i}\right)^{k/r} |\psi_k\rangle$$

$$\theta = k/r.$$

If we apply on ψ_1

$$\theta = 1/r$$

DREAM: Run Phase Estimation $(M_a, |\psi_1\rangle)$

we get $1/r$

→ We have M_a, but we may not have $|\psi_1\rangle$

$$|\psi_1\rangle = \frac{1}{\sqrt{r}}\left(|1\rangle + \omega^{-1}|a\rangle + \omega^{-2}|a^2\rangle + \dots + \omega^{-(r-1)}|a^{r-1}\rangle\right)$$

↳ We don't know r.

$$\frac{1}{\sqrt{r}} \underbrace{\sum_{k=0}^{r-1} |\psi_k\rangle}_{\text{orthogonal}} = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \frac{1}{\sqrt{r}} \left(\sum_{l=0}^{r-1} \omega^{-lk} |a^l\rangle\right)$$

$$= \frac{1}{r} \sum_{l=0}^{r-1} \sum_{k=0}^{r-1} \left(\omega^{-l}\right)^k |a^l\rangle$$

if $l=0$

$$\frac{1}{r}\sum_{k=0}^{r-1}(\omega^0)|a^0\rangle$$

$$\frac{1}{r}\times r\,|1\rangle$$

$$|1\rangle+\frac{1}{r}\sum_{l=1}^{r-1}\underbrace{\sum_{k=0}^{r-1}(\omega^{-l})^k}_{\text{GEOMETRIC SERIES}}|a^l\rangle$$

$$=|1\rangle+\underbrace{\frac{1}{r}\sum_{l=1}^{r-1}\frac{1-(\omega^{-l})^r}{1-(\omega^{-l})}|a^l\rangle}_{0}$$

$$(\omega^{-l})^r=(\omega^r)^{-l}=1$$

$$=|1\rangle$$

SOLUTION TO DREAM PROBLEM:

sum of eigenvectors

$$\underbrace{\text{Run Phase Estimation } (M_a\,|1\rangle)}_{\text{akin to running on a random } |\psi_k\rangle}$$

We get some $\theta = k/r$.

So each time we get k/r

like $k_1/r, k_2/r, k_3/r, \dots, k_p/r$

CONTINUED FRACTION ALGORITHM:

INPUT: $k_1/r, k_2/r, k_3/r, \dots, k_p/r$

OUTPUT: r

USES QUANTUM FOURIER TRANSFORM

SHOR'S ALGORITHM → PHASE ESTIMATION (SUM OF EIGENVECTORS) → CONTINUED FRACTION ALGORITHM

REPEAT

r

CLASSICAL FOURIER TRANSFORM → $N \log N$

QUANTUM FOURIER TRANSFORM → N

Integer Factorization

→ Shor's Algorithm

→ Order Finding

→ Phase Estimation Repeat

→ Quantum Fourier Transform

→ Continued Fraction