2.2.3. Multiple solutions

If there are $r > 1$ marked states and r is known, we can still speed up the search.

(We will discuss later on how to find r if it is not known
$\rightarrow$ phase estimation algorithm).

$$|\tilde{\tau}\rangle = \frac{1}{\sqrt{r}} \sum_{i=1}^{r} |\tau_i\rangle$$: linear superposition of marked (target) states

rotation operator $\hat{Q} = -\hat{I}_\gamma \hat{U}^{-1} \hat{I}_{\tilde{\tau}} \hat{U}$

$$\hat{I}_{\tilde{\tau}} = \hat{I} - 2\sum_{i=1}^{r} |\tau_i\rangle\langle\tau_i|$$

$\hat{Q}$ rotates the vector in the 2-D vector space spanned by $|\gamma\rangle$ and $\hat{U}^{-1}|\tilde{\tau}\rangle$ by θ, where

$$\sin\theta \sim \theta \sim 2\sqrt{\frac{r}{2^n}}$$

The initial state $|\gamma\rangle$ is transferred to the target state $\hat{U}^{-1}|\tilde{\tau}\rangle$ after a number of iterations

$$\frac{\pi}{2\cdot\theta} = \frac{\pi}{4}\sqrt{\frac{2^n}{r}} \longleftarrow$$ In order to truncate the iteration at the optimum point, we must know r.

$\Downarrow$

projective measurement $\Rightarrow$ one of the target states

2.2.4. Geometric picture

$$|\alpha\rangle \equiv \frac{1}{\sqrt{N-r}} \sum_x{}' |x\rangle$$: sum over all x which are not the targets of the search problem.

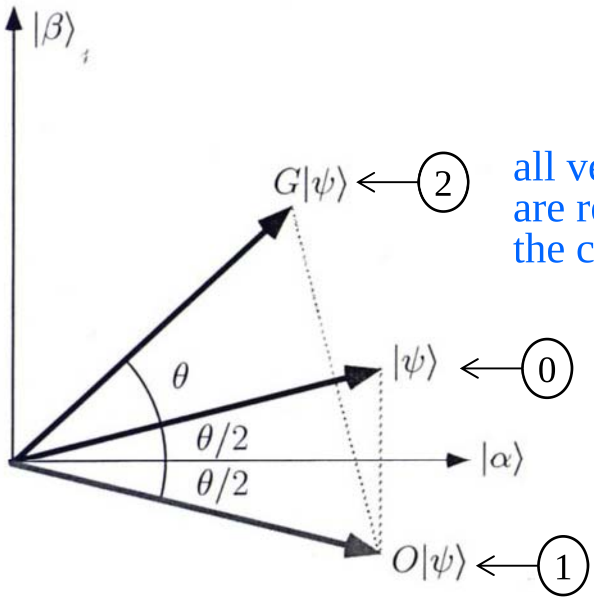
$$|\beta\rangle \equiv \frac{1}{\sqrt{r}} \sum_x{}'' |x\rangle$$: sum over all x which are the targets of the search problem.

$N = 2^n$

initial state $$|\psi\rangle = \sqrt{\frac{N-r}{N}}|\alpha\rangle + \sqrt{\frac{r}{N}}|\beta\rangle = \cos\frac{\theta}{2}|\alpha\rangle + \sin\frac{\theta}{2}|\beta\rangle$$

(after the first W-H gate)

$|\alpha\rangle$ and $|\beta\rangle$ are now exactly orthogonal.



all vectors in $O|\psi\rangle$ are reflected with respect to the corresponding vectors in $|\psi\rangle$

initial state

$|\alpha\rangle$ is preserved but
$|\beta\rangle$ is flipped

Grover oracle = reflection about the vector $|\alpha\rangle$

$\times$ reflection about the vector $|\psi\rangle$

= rotation by θ

Continued application of Grover iterations:

$$\hat{Q}^k|\psi\rangle = \cos\left(\frac{2k+1}{2}\theta\right)|\alpha\rangle + \sin\left(\frac{2k+1}{2}\theta\right)|\beta\rangle$$

optimum iteration: $$\frac{2k+1}{2}\theta \simeq \frac{\pi}{2}$$ $\Rightarrow$ target state $|\beta\rangle$

2.2.5. Grover algorithm is optimum

Let's consider the search problem with a single solution (target) x.

$\Rightarrow$ oracle $\hat{O}_x = \hat{I} - 2|x\rangle\langle x|$

$$|\psi_k^x\rangle = \hat{U}_k \hat{O}_x \hat{U}_{k-1} \hat{O}_x \cdots\cdots \hat{U}_1 \hat{O}_x |\psi_0\rangle$$

initial state

unitary operation

$$|\psi_k\rangle = \hat{U}_k \hat{U}_{k-1} \cdots\cdots \hat{U}_1 |\psi_0\rangle$$

without oracle operation

Measure of the deviation after *k* steps: $D_k = \sum_x ||\psi_k^x - \psi_k||^2$

$|\psi_k^x\rangle$ $|\psi_k\rangle$

We aim to demonstrate

1) D_k can grow no faster than $O(k^2)$.
2) D_k must be $\Omega(N)$ if the probability of success is high.

$\Downarrow$

"A quantum computer cannot search N items by consulting the oracle fewer than $O\left(\sqrt{N}\right)$ times."

$\Downarrow$

"Grover algorithm is optimum."

proof of 1): $D_k \leq 4k^2$

This is clearly true for $k=0$, where $D_k=0$.

$$D_{k+1} = \sum_x \|\hat{O}_x \psi_k^x - \psi_k\|^2 \qquad (\ \hat{U}_{k+1} \text{ does not change } D_{k+1}\ )$$

$$= \sum_x \|\hat{O}_x \left(\psi_k^x - \psi_k\right) + \left(\hat{O}_x - \hat{I}\right)\psi_k\|^2$$

$$\Downarrow$$

$$\|b + c\|^2 \leq \|b\|^2 + 2\|b\|\|c\| + \|c\|^2$$

$$\left(\hat{O}_x - \hat{I}\right)\psi_k = -2\langle x|\psi_k\rangle|x\rangle$$

$$\hat{O}_x \left(\psi_k^x - \psi_k\right)$$

$$\Downarrow$$

$$D_{k+1} \leq \sum_x \left(\|\psi_k^x - \psi_k\|^2 + 4\|\psi_k^x - \psi_k\||\langle x|\psi_k\rangle| + 4|\langle\psi_k|x\rangle|^2\right)$$

(Cauchy-Schwarz inequality for 2nd term)

$$\leq D_k + 4\left(\sum_x \|\psi_k^x - \psi_k\|^2\right)^{1/2}\left(\sum_{x'} |\langle\psi_k|x'\rangle|^2\right)^{1/2} + 4$$

$$\leq D_k + 4\sqrt{D_k} + 4$$

$$\Downarrow$$

If $D_k \leq 4k^2$, then

$$D_{k+1} \leq 4k^2 + 8k + 4 = 4(k+1)^2$$

$$\Downarrow$$

The inductive proof is completed.

proof of 2): $|\langle x|\psi_k^x\rangle|^2 > 1/2$ for all *x* so that the success probability $> 1/2$

Replacing $|x\rangle$ by $e^{i\theta}|x\rangle$ does not change the success probability.

$$\Downarrow$$

$\langle x|\psi_k^x\rangle = |\langle x|\psi_k^x\rangle|$ without loss of generality

$$\Downarrow$$

$$\|\psi_k^x - x\|^2 = 2 - 2|x|\psi_k^x\rangle| \le 2 - \sqrt{2}$$

Define $E_k \equiv \sum_x \|\psi_k^x - x\|^2$, then $E_k \le \left(2-\sqrt{2}\right) N$

Define $F_k \equiv \sum_x \|x - \psi_k^x\|^2$, then $F_k \ge 2N - 2\sqrt{N}$

$$\uparrow$$

$$\left\{\left(1-\frac{1}{\sqrt{N}}\right)^2 + (N-1)\times\frac{1}{N}\right\}\times N$$

$$\Downarrow$$

$$\begin{aligned} D_k &= \sum_x \|\,(\psi_k^x - x) + (x - \psi_k)\,\|^2 \\ &\ge \sum_x \|\psi_k^x - x\|^2 - 2\sum_x \|\psi_k^x - x\|\|x - \psi_k\| + \sum_x \|x - \psi_k\|^2 \\ &= E_k + F_k - 2\sum_x \|\psi_k^x - x\|\|x - \psi_k\| \end{aligned}$$

$$\Downarrow$$

Cauchy-Schwarz inequality

$$\sum_x \|\psi_k^x - x\|\|x - \psi_k\| \le \sqrt{E_k F_k}$$

$$D_k \ge E_k + F_k - 2\sqrt{E_k F_k} = \left(\sqrt{E_k} - \sqrt{F_k}\right)^2$$

$$\Downarrow$$

$D_k \geq C \cdot N$ for sufficiently large N

constant less than $\left(\sqrt{2} - \sqrt{2 - \sqrt{2}}\right)^2 \simeq 0.42$

$$\Downarrow$$

Since $D_k \leq 4k^2$, k must be greater than $k \geq \sqrt{\frac{c}{4} N}$

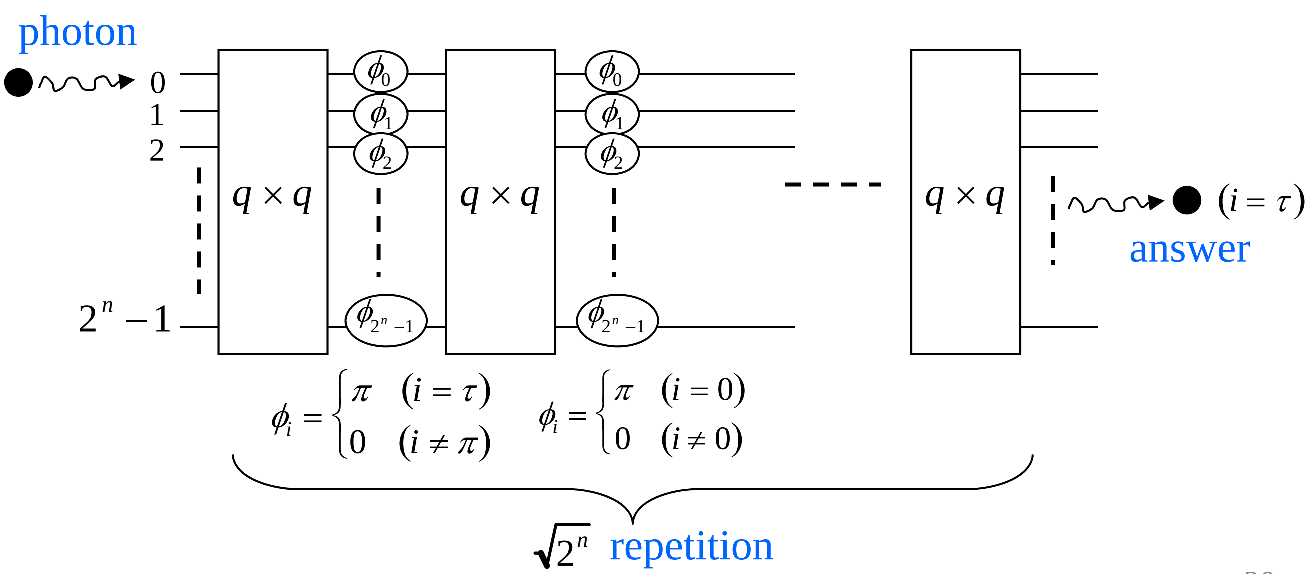
$$\Downarrow$$

To achieve a probability of success $\geq 1/2$ for finding a solution in the search problem, we must call the oracle $\Omega\left(\sqrt{N}\right)$ times.

$$\Downarrow$$

No further improvement is allowed by quantum mechanics. This is because there is no (hidden) structure in the Grever search problem.

2.2.6. A single photon interferometer for implementing Grover algorithm



2.3 Quantum Fourier transform

discrete Fourier transform

$$\underbrace{y_k}_{\text{output }(y_0,y_1,\cdots y_{N-1})} = \frac{1}{\sqrt{N}}\sum_{l=0}^{N-1} \underbrace{x_l}_{\text{input }(x_0,x_1,\cdots x_{N-1})} e^{i\frac{2\pi kl}{N}} \qquad (N=2^n)$$

quantum Fourier transform

$$|l\rangle \xrightarrow{\hat{F}} \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{i\frac{2\pi kl}{N}} |k\rangle$$

Linear superposition $\Rightarrow$ Simultaneous calculation of all y_k s.

$$\underbrace{\sum_{l=0}^{N-1} x_l |l\rangle}_{\text{input}} \xrightarrow{\hat{F}} \underbrace{\sum_{k=0}^{N-1} y_k |k\rangle}_{\text{output}}$$

notation for $N=2^n$ (n qubit case):

$$l = l_1 l_2 \cdots\cdots l_n \longrightarrow l = l_1 2^{n-1} + l_2 2^{n-2} + \cdots\cdots + l_n 2^0$$

similarly, $l = 0.\, l_m l_{m+1} \cdots\cdots l_p = \dfrac{l_m}{2} + \dfrac{l_{m+1}}{2^2} + \cdots\cdots + \dfrac{l_p}{2^{p-m+1}}$

quantum Fourier transform (2)

$$|l\rangle \longrightarrow \frac{1}{\sqrt{2^n}}\sum_{k=0}^{2^n-1} e^{i\frac{2\pi kl}{2^n}} |k\rangle$$

$$\leftarrow k = k_1 k_2 \cdots k_n = k_1 2^{n-1} + k_2 2^{n-2} + \cdots + k_n$$

$$= \frac{1}{\sqrt{2^n}} \underbrace{\sum_{k_1=0}^{1}\sum_{k_2=0}^{1}\cdots\cdots\sum_{k_n=0}^{1}}_{\sum_{k=0}^{2^n-1}} \underbrace{e^{i2\pi l \sum_{j=1}^{n} k_j 2^{-j}}}_{\bigotimes_{j=1}^{n} e^{i2\pi l k_j 2^{-j}}} \underbrace{|k_1 k_2 \cdots\cdots k_n\rangle}_{\bigotimes_{j=1}^{n} |k_j\rangle}$$

$$= \frac{1}{\sqrt{2^n}} \sum_{k_1} \cdots\cdots \sum_{k_n} \bigotimes_{j=1}^{n} e^{i2\pi l k_j 2^{-j}} \left| k_j \right\rangle$$

$$= \frac{1}{\sqrt{2^n}} \bigotimes_{j=1}^{n} \left[\sum_{k_j=0}^{1} e^{i2\pi l k_j 2^{-j}} \left| k_j \right\rangle \right]$$

$$= \frac{1}{\sqrt{2^n}} \bigotimes_{j=1}^{n} \left[|0\rangle + e^{i2\pi l 2^{-j}} |1\rangle \right]$$

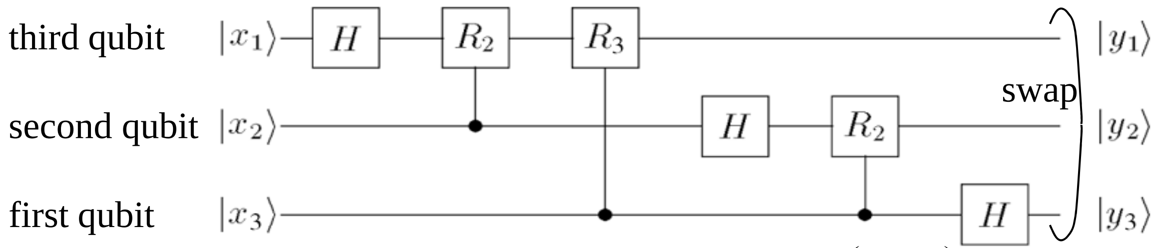
$$= \frac{1}{\sqrt{2^n}} \left[\left(|0\rangle + e^{i2\pi 0.l_n}|1\rangle\right)\left(|0\rangle + e^{i2\pi 0.l_{n-1}l_n}|1\rangle\right) \cdot \cdots\cdots \left(|0\rangle + e^{i2\pi 0.l_1 l_2 \cdots l_n}|1\rangle\right) \right]$$

Example: $j = 1$

$$\frac{l}{2^j} = \underbrace{l_1 2^{n-2} + \cdots + l_{n-1}}_{\text{do not contribute to the phase factor (multiple of } 2\pi)} + \frac{l_n}{2} \Longrightarrow 0.\, l_n$$

Circuit Implementation:

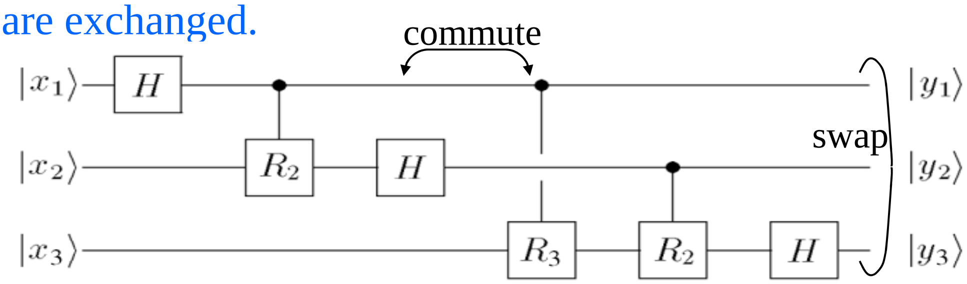
$\boxed{N = 3 \text{ case}}$



$$R_2 = \text{controlled} - \frac{\pi}{2}\text{rotation} \Longrightarrow \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix}$$

$$R_3 = \text{controlled} - \frac{\pi}{4}\text{rotation} \Longrightarrow \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{4}} \end{pmatrix}$$

Controlled-phase shift gate is invariant if control and target qubits are exchanged.



In order to construct the quantum Fourier transform gate, we need a Walsh-Hadamard gate and controlled-phase shift gate:

$$H_i = \frac{1}{\sqrt{2}} \begin{array}{c} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{array} \begin{array}{l} |0\rangle_i \\ |1\rangle_i \end{array}$$

computational basis

$$R_l = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & e^{i\theta} \end{bmatrix} \begin{array}{l} |0\rangle_j |0\rangle_k \\ |0\rangle_j |1\rangle_k \\ |1\rangle_j |0\rangle_k \\ |1\rangle_j |1\rangle_k \end{array}$$

$$\theta = \frac{\pi}{2^l}$$

computational basis

$$R_2 \quad (l=1)$$

$$R_3 \quad (l=2)$$

$$\vdots$$

input state: $| j_1 \cdots j_n \rangle \xrightarrow{\hat{H}_1} \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_1} |1\rangle \right) | j_2 \cdots j_n \rangle$

$$e^{i2\pi 0.j_1} = \begin{cases} 1 : j_1 = 0 \\ -1 : j_1 = 1 \end{cases}$$

$$\xrightarrow{\hat{R}_2} \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_1 j_2} |1\rangle \right) | j_2 \cdots j_n \rangle$$

$$\xrightarrow{\hat{R}_3 \hat{R}_4 \cdots \hat{R}_n} \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2\pi 0.j_1 j_2 \cdots j_n} |1\rangle \right) | j_2 \cdots j_n \rangle$$

$$\xrightarrow{\hat{H}_2} \frac{1}{\sqrt{2^2}} \left(|0\rangle + e^{i2\pi 0.j_1 j_2 \cdots j_n} |1\rangle \right) \left(|0\rangle + e^{i2\pi 0.j_2} |1\rangle \right) | j_3 \cdots j_n \rangle$$

$$\xrightarrow{\hat{R}_2 \cdots \hat{R}_{n-1}} \frac{1}{\sqrt{2^2}} \left(|0\rangle + e^{i2\pi 0.j_1 j_2 \cdots j_n} |1\rangle \right) \left(|0\rangle + e^{i2\pi 0.j_2 \cdots j_n} |1\rangle \right) | j_3 \cdots j_n \rangle$$

$\Downarrow$ continuation of the similar operations

$$\frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{i2\pi 0.j_1 j_2 \cdots j_n}|1\rangle\right)\left(|0\rangle + e^{i2\pi 0.j_2 \cdots j_n}|1\rangle\right)\cdots$$

$$\cdots\left(|0\rangle + e^{i2\pi 0.j_n}|1\rangle\right)$$

$\Downarrow$ swap operation

$$\frac{1}{\sqrt{2^n}}\left(|0\rangle + e^{i2\pi 0.j_n}|1\rangle\right)\left(|0\rangle + e^{i2\pi 0.j_{n-1} j_n}|1\rangle\right)\cdots$$

$$\cdots\left(|0\rangle + e^{i2\pi 0.j_1 j_2 \cdots j_n}|1\rangle\right)$$

Appendix. Number theoretic preparation for Shor's algorithm

Factoring algorithm

1 Choose a positive integer number x randomly which is smaller than N and relatively prime to N. Find an order r defined by the relation:

$$x^r \equiv 1 \,(\mathrm{mod}\, N) \quad \Rightarrow \quad x^r = p \cdot N + 1$$

smallest positive integer positive integer

2 If r is an even number,

$$\left(x^{\frac{r}{2}}+1\right)\left(x^{\frac{r}{2}}-1\right) = pN$$, and also

if $x^{\frac{r}{2}} \pm 1 \neq 0 \,(\mathrm{mod}\, N)$, $\gcd\left(x^{\frac{r}{2}}+1, N\right)$ or $\gcd\left(x^{\frac{r}{2}}-1, N\right)$ provides the factors of N.

greatest common divisor

3 If r is an odd number or $x^{\frac{r}{2}} \pm 1 = 0 \,(\mathrm{mod}\, N)$, $\gcd\left(x^{\frac{r}{2}}+1, N\right)$ and $\gcd\left(x^{\frac{r}{2}}-1, N\right)$ provide the trivial factors 1 and N.

$\Rightarrow$ try another positive integer number x

$\Downarrow$

The probability of finding a desired r for a randomly chosen x is greater than 50%. (Chinese reminder theorem)

$$1 - \frac{1}{2^{k-1}} > \frac{1}{2}$$

k: # of distinct odd primes of N

Example: $N = 15$, $x = \{2, 4, 7, 8, 11, 13, 14\}$

i) $x = 2$

$x^4 = 16 = 15 + 1$

$r = 4$ (order)

$x^{r/2} - 1 = 3 \longrightarrow \gcd(3, 15) = 3$

$x^{r/2} + 1 = 5 \longrightarrow \gcd(5, 15) = 5$

ii) $x = 4$

$x^2 = 16 = 15 + 1$

$r = 2$ (order)

$x^{r/2} - 1 = 3$

$x^{r/2} + 1 = 5$

iii) $x = 7$

$x^4 = 2401 = 15 \text{ x } 60 + 1$

$r = 4$ (order)

$x^{r/2} - 1 = 48 \longrightarrow \gcd(48, 15) = 3$

$x^{r/2} + 1 = 50 \longrightarrow \gcd(50, 15) = 5$

iv) $x = 8$

$x^4 = 4096 = 15 \text{ x } 273 + 1$

$r = 4$ (order)

$x^{r/2} - 1 = 63 \longrightarrow \gcd(63, 15) = 3$

$x^{r/2} + 1 = 65 \longrightarrow \gcd(65, 15) = 5$

v) $x = 11$

$x^2 = 121 = 15 \text{ x } 8 + 1$

$r = 2$ (order)

$x^{r/2} - 1 = 10 \longrightarrow \gcd(10, 15) = 3$

$x^{r/2} + 1 = 12 \longrightarrow \gcd(12, 15) = 5$

vi) $x = 13$ $\quad x^4 = 28561 = 15 \text{ x } 1904 + 1$

$r = 4$ (order)

$x^{r/2} - 1 = 168 \longrightarrow$ gcd (168, 15) = 3

$x^{r/2} + 1 = 170 \longrightarrow$ gcd (170, 15) = 5

vii) $x = 14$ $\quad x^2 = 196 = 15 \text{ x } 13 + 1$

$r = 2$ (order)

$x^{r/2} - 1 = 13$

$x^{r/2} + 1 = 15$

$\Rightarrow \times$

gcd (13, 15) = 1

gcd (15, 15) = 15

Euclidean algorithm for gcd ($x^{r/2} \pm 1, N$)

gcd (a, b) $\quad$ ($a > b$)

(1) Compare a-b and b, subtract a small number from a large number

(2) Repeat the process.

(3) The number just before the final result (zero) is a desired result gcd (a, b).

Example: gcd (20, 12)

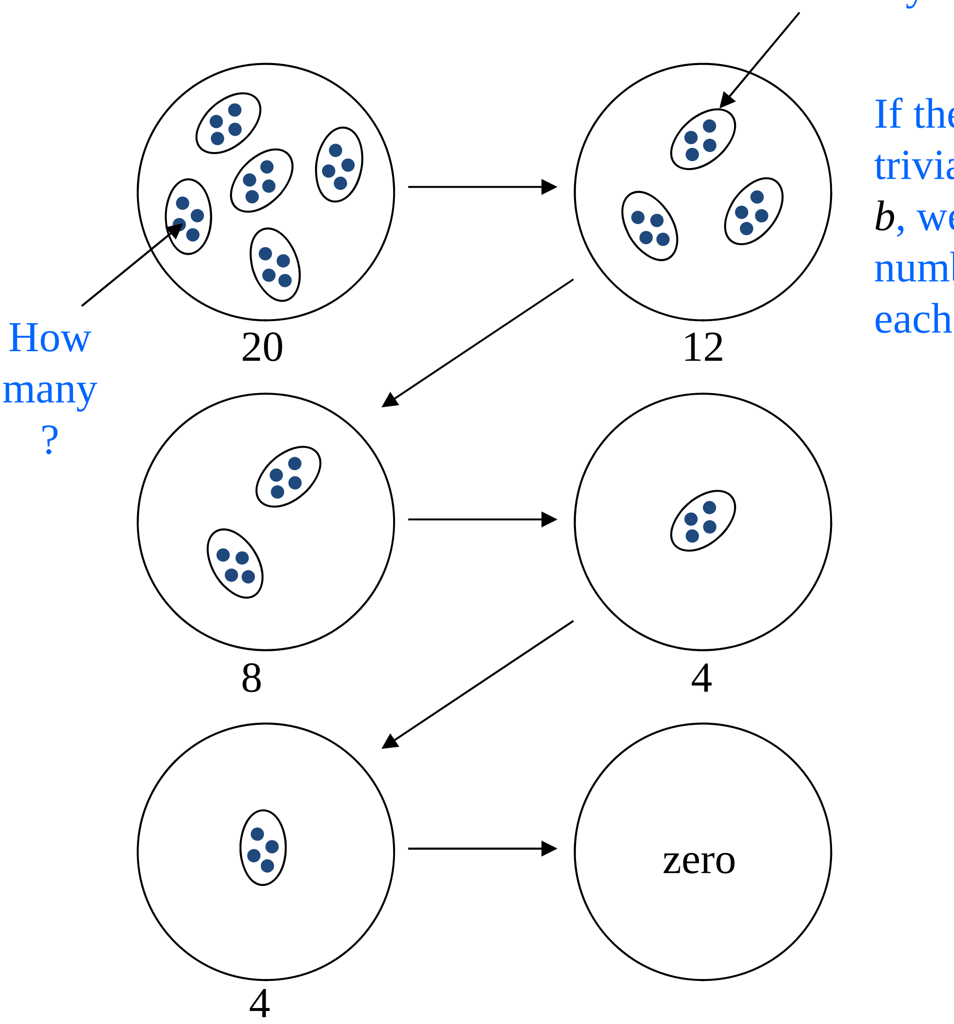
a $\quad$ b

20 $\longrightarrow$ 12

8 $\longrightarrow$ 4

desired result $\Rightarrow$ 4 $\longrightarrow$ 0 $\Leftarrow$ final result

Interpretation:



If there is a non-trivial gcd for a and b, we can put this number of coins in each packet.

Finding a gcd $(x^{r/2} \pm 1, N)$ takes $\sim(\log N)^3$ computation steps.

⇨ polynomial time

The difficulty of factoring a large compound number N is the step of finding an order r.