9-2 Grover's Search Algorithm

§6.1 Quantum search algorithm

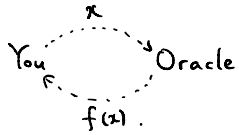
💬 polynomial speed-up ($N \to \sqrt{N}$)

objective $x \in \{1, 2, \ldots, N\}$, solutions $S = \{s_1, s_2, \ldots, s_M\}$, $M \ll N$

$$f(x) = \begin{cases} 1, & x \in S \\ 0, & \text{otherwise} \end{cases}$$

For given x, $f(x)$ is easy to calculate.

Find S.



Classically, you need $O(N)$ oracle queries.

💬 $O(N)$: order of N (big-O notation)

💬 Polynomial vs Non-polynomial

poly$(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k$, k: finite

Non-polynomial: super-polynomial, exponential, ⋯

n: size of the problem (~ # of bits)

In complexity theory,

if you need poly(n) time steps to do something, it is manageable.

if you need non-poly(n) time steps to do something, it is hopeless.

💬 Reduction (환원)

Problem A $\xrightarrow{\text{reduction in poly}(n)\text{ steps}}$ Problem B

Problem B is at least as hard as Problem A.

Possibly harder.

💬 P (Polynomial time) vs NP (Non-deterministic Polynomial time) (complexity classes)

Problems in class P (P problems):

solutions can be found in polynomial time in a classical computer.

Problems in class NP (NP problems):

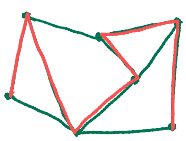
given a solution, its validity can be checked in polynomial time.

Apparently, P $\subset$ NP.

P $\overset{?}{\neq}$ NP: one of the seven (now six) Millennium Prize Problems (US$1m, Clay institute)

💬 NP problems

ex) Hamiltonian paths (travelling salesman problems)



ex) satisfiability problems

$(\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (x_1 \vee x_3) \wedge (x_3 \vee x_2)$ (2-SAT problem)

$\neg$: NOT, $\vee$: OR, $\wedge$: AND

solution: 001

Any NP problem $\xrightarrow{\text{reduction in poly}(n)\text{ steps}}$ Any NP-complete problem

meaning: if you can solve one NP-complete problem, you can solve all NP problems!

NP-complete problems: 3-SAT, Hamiltonian paths, ···

BQP (Bounded-error Quantum Polynomial time)

efficiently solvable in a quantum computer up to a bounded probability of error

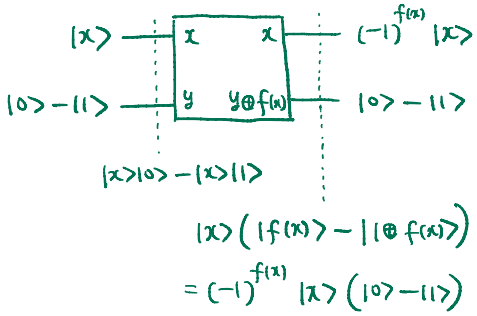
quantum version of BPP (Bounded-error Probabilistic Polynomial time, $\text{BPP} \stackrel{?}{=} \text{P}$)

google search "complexity zoo", "quantum algorithm zoo"

algorithm

oracle operation O: $|x\rangle \to -|x\rangle$ if x is a solution

$|x\rangle \to |x\rangle$ otherwise



$$|\psi\rangle \equiv \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle = \frac{1}{\sqrt{2^n}} \sum_{x_1,x_2,\ldots,x_n \in \{0,1\}} |x_1 x_2 \ldots x_n\rangle = H \otimes H \otimes \cdots \otimes H |00\ldots0\rangle = H^{\otimes n}|0\rangle$$

Grover iteration G: ① oracle operation O

② $H^{\otimes n}$

③ $|0\rangle \to |0\rangle$, $|x\rangle \to -|x\rangle$ for all $x \neq 0$ $(= |0\rangle \to -|0\rangle,\ |x\rangle \to |x\rangle\ \ \forall x \neq 0)$

④ $H^{\otimes n}$

③: $|0\rangle\langle 0| - |1\rangle\langle 1| - |2\rangle\langle 2| - \cdots - |N-1\rangle\langle N-1| = 2|0\rangle\langle 0| - I$

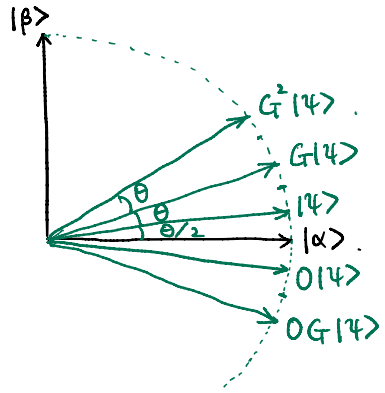
②~④: $H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} = 2|\psi\rangle\langle\psi| - I$

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

how it works

$$|\alpha\rangle \equiv \frac{1}{\sqrt{N-M}} \sum_{x \notin S} |x\rangle, \quad |\beta\rangle \equiv \frac{1}{\sqrt{M}} \sum_{x \in S} |x\rangle \qquad M\text{: \# of solutions}$$

$$|\psi\rangle = \sqrt{1 - \frac{M}{N}}|\alpha\rangle + \sqrt{\frac{M}{N}}|\beta\rangle \equiv \cos\frac{\theta}{2}|\alpha\rangle + \sin\frac{\theta}{2}|\beta\rangle$$



O: $|\alpha\rangle \to |\alpha\rangle, \quad |\beta\rangle \to -|\beta\rangle$

$\Rightarrow$ inversion about $|\alpha\rangle$

$2|\psi\rangle\langle\psi| - I$: $\langle\psi|(2|\psi\rangle\langle\psi| - I)|\phi\rangle = 2\langle\psi|\phi\rangle - \langle\psi|\phi\rangle = \langle\psi|\phi\rangle$

$\Rightarrow$ angle between $|\psi\rangle$ and $(2|\psi\rangle\langle\psi| - I)|\phi\rangle$

= angle between $|\psi\rangle$ and $|\phi\rangle$

$\Rightarrow$ inversion about $|\psi\rangle$

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

$k \sim \dfrac{\pi/2}{\theta} \sim \sqrt{\dfrac{N}{M}}$ iterations are needed to get close to $|\beta\rangle$

- Explicitly demonstrate each step of the Grover's algorithm for $N = 4$ (two qubits).
- Unfortunately, Grover's algorithm is optimal! (proven)

  No algorithm can have a better-than-$\sqrt{N}$ speedup.