Ch 1 알고리즘 (algorithm)

problem → algorithm
(문제)

컴퓨터로 문제를 해결하는
유한한 단계적 해법

- ① Halt (finite)
- ② correct answer
- ③ input ≥ 0, output ≥ 1

ex) 문제

$$S = \sum_{i=a}^{b} i$$

Algorithm

```
S=0
for(i=a to b by 1)
{ S ← S+i }
print S
```

ex) Fermat's Last Theorem (페르마의 마지막 정리)

"모든 정수 $n>2$에 대해 $x^n+y^n=z^n$을 만족하는 정수 $x, y, z>0$가 존재하는 것이다"

$\neg \forall n>2 \quad \exists x, y, z>0 \quad x^n+y^n=z^n$

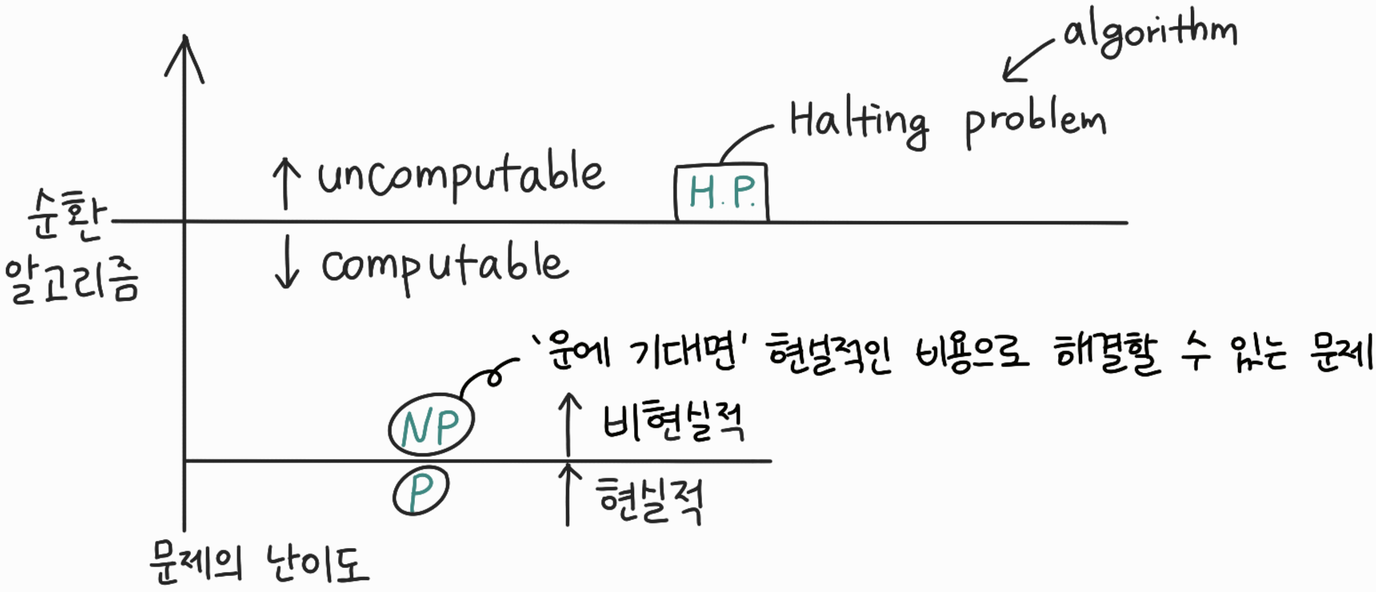
(수학계 300년 미해결 난제, 1995년 증명)

해법 (?)

```
n > 2 (=for (n=3 to ∞)
for (x=1 to ∞)
    for (y=1 to ∞)
        for (z=1 to ∞)
            if(x^n+y^n= z^n) stop
```

algorithm 인가? NO! (끝나지 않는다)

수학적 해법은 될 수 있으나 → procedure



문제 → algorithm → f(n) : 성능함수 (적을수록 좋음)

input size = n

$2n^2 + 4n - 9$

복잡도 (complexity)
- 시간 (time)
- 공간 (space)

* 성능함수의 Big-O 표기법 (Asymtotic Notation)

$n^2 = O(n^2)$

$\frac{1}{100억} n^2 - 10억\, n = O(n^2) = O(n^3)$

$\frac{1}{2}n^2 + n\log n = O(n^2) = o(n^2)$

* 여러가지 표기법

$f(n) = O(g(n)) \Leftrightarrow f(n) \leq g(n)$ g(n)보다 성능이 나쁘지 않다 (하한)

$f(n) = \Omega(g(n)) \Leftrightarrow f(n) \geq g(n)$ 아무리 좋아도 g(n)이다 (상한)

$f(n) = \Theta(g(n)) \Leftrightarrow f(n) = g(n)$

$f(n) = o(g(n)) \Leftrightarrow f(n) < g(n)$

$f(n) = \omega(g(n)) \Leftrightarrow f(n) > g(n)$

ex) $1 < \lg n < \sqrt{n} < n < n\log n < n^2 < n^3 \cdots$ | $\cdots 2^n < n!$

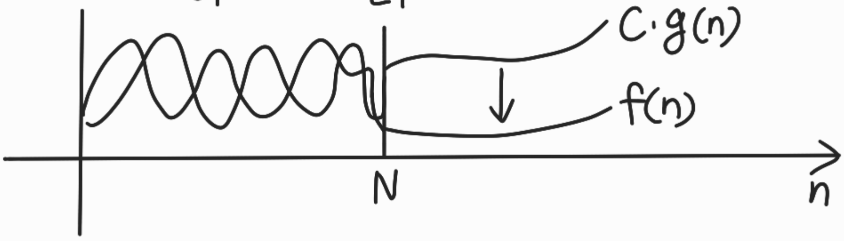
P | NP

$\log_2 x = \lg x$

def) $f(n) = O(g(n)) \iff$

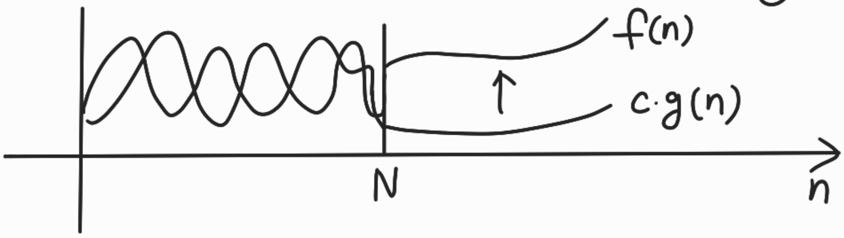
$\forall n \geq N \quad \exists\, c > 0 \quad f(n) \leq c \cdot g(n)$

정수 실수



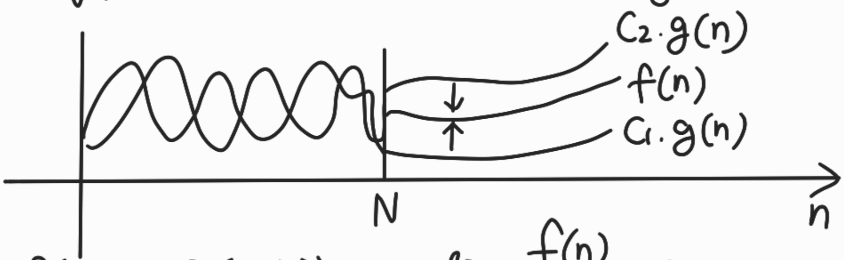
def) $f(n) = \Omega(g(n)) \iff$

$\forall n \geq N \quad \exists\, c > 0 \quad f(n) \geq c \cdot g(n)$



def) $f(n) = \Theta(g(n)) \iff f(n) = O(g(n) \wedge f(n)) = \Omega(g(n)$

$\forall n \geq N \quad \exists\, c_1, c_2 > 0 \quad c_1 g_1(n) \leq f(n) \leq c_2 g(n)$



def) $f(n) = o(g(n)) \iff \lim_{n \to \infty} \frac{f(n)}{g(g)} = 0$

$f(n) = \omega(g(n)) \iff \lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty$