차수(Order)

강의 슬라이드 #1b
도경구

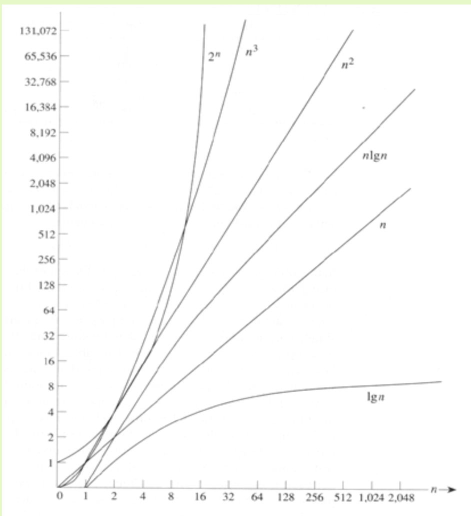
대표적인 복잡도 카테고리

- $\Theta(\lg n)$
- $\Theta(n)$: 1차(linear)
- $\Theta(n \lg n)$
- $\Theta(n^2)$: 2차(quadratic)
- $\Theta(n^3)$: 3차(cubic)
- $\Theta(2^n)$: 지수(exponential)
- $\Theta(n!)$

예제: 2차 항이 궁극적으로 지배한다.

n	$0.1n^2$	$0.1n^2+n+100$
10	10	120
20	40	160
50	250	400
100	1,000	1,200
1,000	100,000	101,100

복잡도 함수의 증가율



시간복잡도별 실행시간 비교

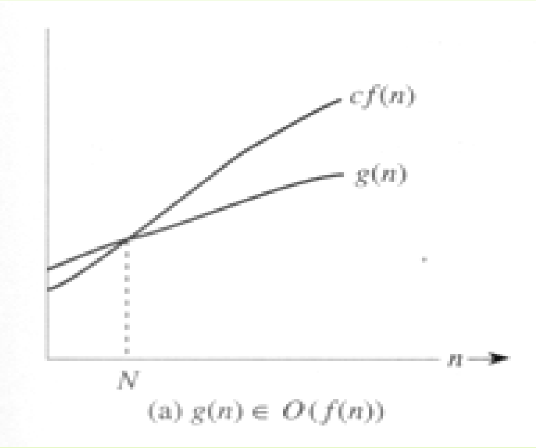
n	$f(n) = \lg n$	$f(n) = n$	$f(n) = n \lg n$	$f(n) = n^2$	$f(n) = n^3$	$f(n) = 2^n$
10	0.003 μs*	0.01 μs	0.033 μs	0.1 μs	1 μs	1 μs
20	0.004 μs	0.02 μs	0.086 μs	0.4 μs	8 μs	1 ms†
30	0.005 μs	0.03 μs	0.147 μs	0.9 μs	27 μs	1 s
40	0.005 μs	0.04 μs	0.213 μs	1.6 μs	64 μs	18.3 min
50	0.006 μs	0.05 μs	0.282 μs	2.5 μs	125 μs	13 days
10^2	0.007 μs	0.10 μs	0.664 μs	10 μs	1 ms	4×10^{13} years
10^3	0.010 μs	1.00 μs	9.966 μs	1 ms	1 s	
10^4	0.013 μs	10 μs	130 μs	100 ms	16.7 min	
10^5	0.017 μs	0.10 ms	1.67 ms	10 s	11.6 days	
10^6	0.020 μs	1 ms	19.93 ms	16.7 min	31.7 days	
10^7	0.023 μs	0.01 s	0.23 s	1.16 days	31,709 years	
10^8	0.027 μs	0.10 s	2.66 s	115.7 days	3.17×10^7 years	
10^9	0.030 μs	1 s	29.90 s	31.7 days		

* 1 μs = 10^{-6} second
† 1 ms = 10^{-3} second

큰(Big)O 표기법

- **정의 : 점근적 상한(Asymptotic Upper Bound)**
  - ✓ 주어진 복잡도 함수 $f(n)$에 대해서 $g(n) \in O(f(n))$ 이면 다음을 만족한다.
  - ✓ $n \geq N$인 모든 정수 n에 대해서 $g(n) \leq c \times f(n)$이 성립하는 실수 $c > 0$와 음이 아닌 정수 N이 존재한다.
- $g(n) \in O(f(n))$ 읽는 방법:
  - ✓ $g(n)$은 $f(n)$의 큰 오(big O)

큰 O 표기법



2001 -03 -19 알고리즘 강의 슬라이드 1b 7

큰 O 표기법

- 어떤 함수 $g(n)$이 $O(n^2)$에 속한다는 말은
  - 그 함수는 궁극에 가서는 (즉, 어떤 임의의 N값보다 큰 값에 대해서는) 어떤 2차 함수 cn^2 보다는 **작은** 값을 가지게 된다는 것을 뜻한다.
    (그래프 상에서는 **아래**에 위치)
  - 다시 말해서, 그 함수 $g(n)$은 어떤 2차 함수 cn^2 보다는 궁극적으로 **좋다**고 (기울기가 **낮다**고) 말할 수 있다.
- 어떤 알고리즘의 시간복잡도가 $O(f(n))$이라면
  - 입력의 크기 n에 대해서 이 알고리즘의 수행시간은 **아무리 늦어도** $f(n)$은 된다. ($f(n)$이 **상한**이다.)
  - 다시 말하면, 이 알고리즘의 수행시간은 $f(n)$보다 절대로 더 느릴 수는 없다는 말이다.

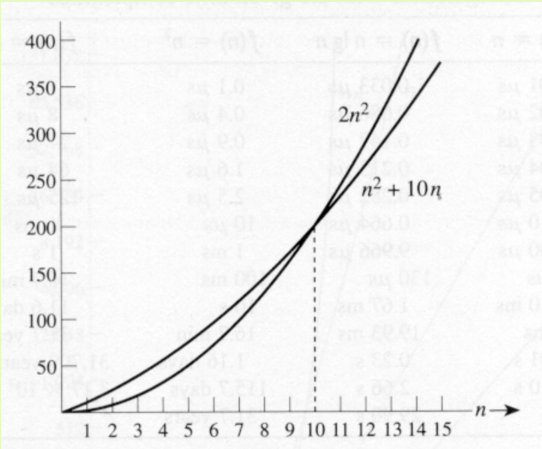
2001 -03 -19 알고리즘 강의 슬라이드 1b 8

큰 O 표기법 : 예

- $n^2+10n \in O(n^2)$?
  - (1) $n \geq 10$인 모든 정수 n에 대해서 $n^2+10n \leq 2n^2$ 이 성립한다. 그러므로, $c = 2$와 $N = 10$을 선택하면, "큰 O"의 정의에 의해서 $n^2+10n \in O(n^2)$이라고 결론지을 수 있다.
  - (2) $n \geq 1$인 모든 정수 n에 대해서 $n^2+10n \leq n^2+10n = 11n^2$ 이 성립한다. 그러므로, $c = 11$와 $N = 1$을 선택하면, "큰 O"의 정의에 의해서 $n^2+10n \in O(n^2)$ 이라고 결론지을 수 있다.

2001 -03 -19 알고리즘 강의 슬라이드 1b 9

$2n^2$과 $n^2 + 10n$의 비교



2001 -03 -19 알고리즘 강의 슬라이드 1b 10

큰 O 표기법 : 예 (계속)

- $5n^2 \in O(n^2)$?

  c=5와 N=0을 선택하면, $n \geq 0$인 모든 정수 n에 대해서 $5n^2 \leq 5n^2$ 이 성립한다.

- $T(n) = \frac{n(n-1)}{2}$?

  $n \geq 0$인 모든 정수 n에 대해서 $\frac{n(n-1)}{2} \leq \frac{n^2}{2}$ 이 성립한다. 그러므로, $c = \frac{1}{2}$ 와 N=0을 선택하면, $T(n) \in O(n^2)$이라고 결론지을 수 있다.

- $n^2 \in O(n^2+10n)$?

  $n \geq 0$인 모든 정수 n에 대해서, $n^2 \leq 1 \times (n^2+10n)$이 성립한다. 그러므로, c=1와 N=0을 선택하면, $n^2 \in O(n^2+10n)$이라고 결론지을 수 있다.

2001 -03 -19 알고리즘 강의 슬라이드 1b 11

큰 O 표기법 : 예 (계속)

- $n \in O(n^2)$?

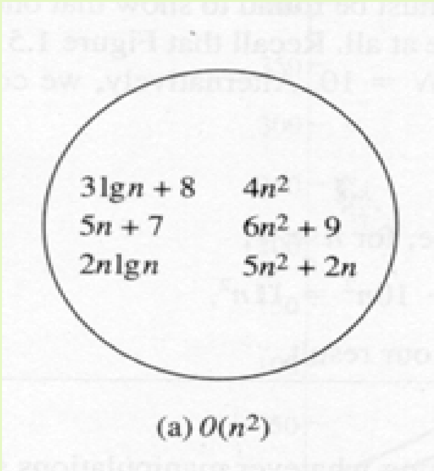
  $n \geq 1$인 모든 정수 n에 대해서, $n \leq 1 \times n^2$ 이 성립한다. 그러므로, c=1와 N=1을 선택하면, $n \in O(n^2)$이라고 결론지을 수 있다.

- $n^3 \in O(n^2)$?

  $n \geq N$인 모든 n에 대해서 $n^3 \leq c \times n^2$ 이 성립하는 c와 N 값은 존재하지 않는다. 즉, 양변을 n^2으로 나누면, $n \leq c$ 가 되는데 c를 아무리 크게 잡더라도 그 보다 더 큰 n이 존재한다.

2001 -03 -19 알고리즘 강의 슬라이드 1b 12

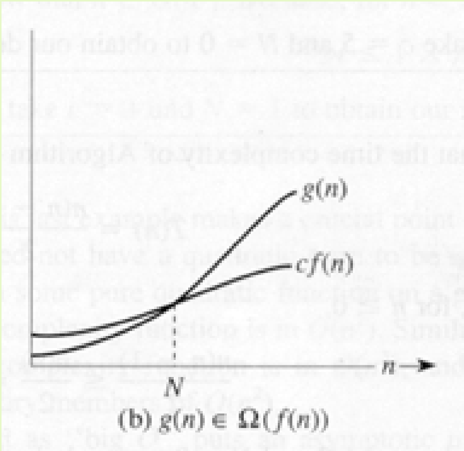
$O(n^2)$



Ω 표기법

- **정의 : 점근적 하한(Asymptotic Lower Bound)**
  - ✓ 주어진 복잡도 함수 $f(n)$에 대해서 $g(n) \in \Omega(f(n))$이면 다음을 만족한다.
  - ✓ $n \geq N$인 모든 정수 n에 대해서 $g(n) \geq c \times f(n)$이 성립하는 실수 $c > 0$와 음이 아닌 정수 N이 존재한다.
- $g(n) \in \Omega(f(n))$ 읽는 방법:
  - $g(n)$은 $f(n)$의 오메가(omega)

Ω 표기법



2001 -03 -19 알고리즘 강의 슬라이드 1b 15

Ω 표기법

- 어떤 함수 $g(n)$이 $\Omega(n^2)$에 속한다는 말은
  - 그 함수는 궁극에 가서는 (즉, 어떤 임의의 N 값보다 큰 값에 대해서는) 어떤 2차 함수 cn^2 의 값보다는 **큰** 값을 가지게 된다는 것을 뜻한다.
    (그래프 상에서는 **위**에 위치)
  - 다시 말해서, 그 함수 $g(n)$은 어떤 2차 함수 cn^2 보다는 궁극적으로 **나쁘다**고 (기울기가 **높다**고)말할 수 있다.
- 어떤 알고리즘의 시간복잡도가 $\Omega(f(n))$이라면,
  - 입력의 크기 n에 대해서 이 알고리즘의 수행시간은 **아무리 빨라도** $f(n)$밖에 되지 않는다. ($f(n)$이 **하한**이다.)
  - 다시 말하면, 이 알고리즘의 수행시간은 $f(n)$보다 절대로 더 빠를 수는 없다는 말이다.

2001 -03 -19 알고리즘 강의 슬라이드 1b 16

Ω 표기법 : 예

- $n^2+10n \in \Omega(n^2)$?

  $n \geq 0$인 모든 정수 n에 대해서 $n^2+10n \geq n^2$ 이 성립한다. 그러므로, $c = 1$와 $N = 0$을 선택하면, $n^2+10n \in \Omega(n^2)$이라고 결론지을 수 있다.
- $5n^2 \in \Omega(n^2)$?

  $n \geq 0$인 모든 정수 n에 대해서, $5n^2 \geq 1 \times n^2$ 이 성립한다. 그러므로, $c=1$와 $N=0$을 선택하면, $5n^2 \in \Omega(n^2)$이라고 결론지을 수 있다.

2001 -03 -19 알고리즘 강의 슬라이드 1b 17

Ω 표기법 : 예 (계속)

- $T(n) = \frac{n(n-1)}{2}$?

  $n \geq 2$인 모든 n에 대해서 $n-1 \geq \frac{n}{2}$이 성립한다. 그러므로, $n \geq 2$인 모든 n에 대해서 $\frac{n(n-1)}{2} \geq \frac{n}{2} \times \frac{n}{2} = \frac{1}{4}n^2$ 이 성립한다. 따라서 $c=\frac{1}{4}$과 $N = 2$를 선택하면, $T(n) \in \Omega(n^2)$이라고 결론지을 수 있다.
- $n^3 \in \Omega(n^2)$?

  $n \geq 1$인 모든 정수 n에 대해서, $n^3 \geq 1 \times n^2$이 성립한다. 그러므로, $c = 1$와 $N = 1$을 선택하면, $n^3 \in \Omega(n^2)$이라고 결론지을 수 있다.

2001 -03 -19 알고리즘 강의 슬라이드 1b 18

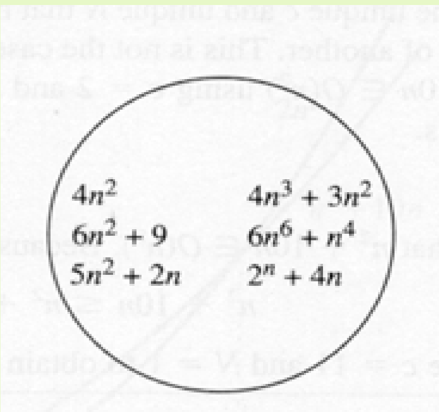
Ω 표기법 : 예 (계속)

- $n \in \Omega(n^2)$?

  <u>모순유도에 의한 증명(Proof by contradiction)</u>:

  $n \in \Omega(n^2)$이라고 가정. 그러면 $n \geq N$인 모든 정수 n에 대해서, $n \geq c \times n^2$이 성립하는 실수 $c > 0$, 그리고 음이 아닌 정수 N이 존재한다. 위의 부등식의 양변을 cn으로 나누면 $\frac{1}{c} \geq n$가 된다. 그러나 이 부등식은 절대로 성립할 수 없다. 따라서 위의 가정은 모순이다.

$\Omega(n^2)$



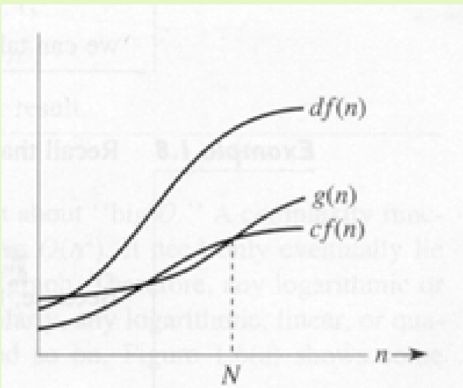
(b) $\Omega(n^2)$

Θ 표기법

- 정의 : **(Asymptotic Tight Bound)**
  - ✓ 복잡도 함수 $f(n)$에 대해서 $\Theta(f(n)) = O(f(n)) \cap \Omega(f(n))$
  - ✓ 다시 말하면, $\Theta(f(n))$은 다음을 만족하는 복잡도 함수 $g(n)$의 집합이다.
  - ✓ 즉, $n \geq N$ 인 모든 정수 n에 대해서 $c \times f(n) \leq g(n) \leq d \times f(n)$이 성립하는 실수 $c>0$와 $d>0$, 그리고 음이 아닌 정수 N이 존재한다.
- 참고: $g(n) \in \Theta(f(n))$은 “$g(n)$은 $f(n)$의 차수(order)”라고 한다.
- 예 : $T(n) = \frac{n(n-1)}{2}$ 은 $O(n^2)$이면서 $\Omega(n^2)$이다. 따라서 $T(n) = \Theta(n^2)$

2001 -03 -19 알고리즘 강의 슬라이드 1b 21

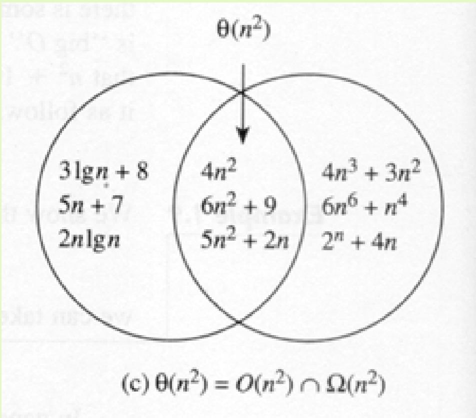
Θ 표기법



(c) $g(n) \in \Theta(f(n))$

2001 -03 -19 알고리즘 강의 슬라이드 1b 22

$\Theta(n^2)$



2001 -03 -19 알고리즘 강의 슬라이드 1b 23

작은(Small) o 표기법

- 작은 o는 복잡도 함수 끼리의 관계를 나타내기 위한 표기법이다.
- **정의 : 작은 o**
  주어진 복잡도 함수 $f(n)$에 대해서 $o(f(n))$ 은 다음을 만족하는 모든 복잡도 함수 $g(n)$의 집합이다: 모든 실수 $c > 0$에 대해서 $g(n) \le c \times f(n)$ (여기서 $n \ge N$ 인 모든 n에 대해서)이 성립하는 음이 아닌 정수 N이 존재한다.
- 참고: $g(n) \in o(f(n))$은 “ $g(n)$은 $f(n)$의 작은 오(o)”라고 한다.

2001 -03 -19 알고리즘 강의 슬라이드 1b 24

큰 O vs.작은 o

- 큰 O와의 차이점
  - 큰 O - 실수 $c>0$ 중에서 하나만 성립하여도 됨
  - 작은 o - 모든 실수 $c>0$에 대해서 성립하여야 함
- $g(n) \in o(f(n))$은 쉽게 설명하자면 $g(n)$이 궁극적으로 $f(n)$보다 훨씬 낫다(좋다)는 의미이다.

2001-03-19 알고리즘 강의 슬라이드 1b 25

작은 o 표기법 : 예

- $n \in o(n^2)$?

증명: $c>0$이라고 하자. $n \geq N$인 모든 n에 대해서 $n \leq cn^2$이 성립하는 N을 찾아야 한다. 이 부등식의 양변을 cn으로 나누면 $\frac{1}{c} \leq n$을 얻는다. 따라서 $N \geq \frac{1}{c}$가 되는 어떤 N을 찾으면 된다. 여기서 N의 값은 c에 의해 좌우된다. 예를 들어 만약 c=0.0001이라고 하면, N의 값은 최소한 10,000이 되어야 한다. 즉, $n \geq 10{,}000$인 모든 n에 대해서 $n \leq 10{,}000n^2$이 성립한다.

2001-03-19 알고리즘 강의 슬라이드 1b 26

작은 o 표기법 : 예 (계속)

- n이 $o(5n)$?
  모순 유도에 의한 증명: $c = \frac{1}{6}$이라고 하자.
  $n \in o(5n)$이라고 가정하면, $n \geq N$인 모든 정수 n에 대해서, $n \leq \frac{1}{6} \times 5n = \frac{5}{6}n$이 성립하는 음이 아닌 정수 N이 존재해야 한다. 그러나 그런 N은 절대로 있을 수 없다. 따라서 위의 가정은 모순이다.

2001 -03 -19 알고리즘 강의 슬라이드 1b 27

차수의 주요 성질 I

- $g(n) \in O(f(n))$ iff $f(n) \in \Omega(g(n))$
- $g(n) \in \Theta(f(n))$ iff $f(n) \in \Theta(g(n))$
- $b > 1$이고 $a > 1$이면, $\log_a n \in \Theta(\log_b n)$은 항상 성립. 다시 말하면 로그(logarithm) 복잡도 함수는 모두 같은 카테고리에 속한다. 따라서 통상 $\Theta(\lg n)$으로 표시한다.
- $b > a > 0$이면, $a^n \in o(b^n)$. 다시 말하면, 지수(exponential) 복잡도 함수가 모두 같은 카테고리 안에 있는 것은 아니다.

2001 -03 -19 알고리즘 강의 슬라이드 1b 28

차수의 주요 성질 II

- $a > 0$인 모든 a에 대해서, $a^n \in o(n!)$. 다시 말하면, $n!$은 어떤 지수 복잡도 함수보다도 나쁘다.
- 복잡도 함수를 다음 순으로 나열해 보자.
  $\Theta(\lg n), \Theta(n), \Theta(n \lg n), \Theta(n^2), \Theta(n^j), \Theta(n^k), \Theta(a^n), \Theta(b^n), \Theta(n!)$
  여기서 $k>j>2$이고 $b>a>1$이다. 복잡도함수 $g(n)$이 $f(n)$을 포함한 카테고리의 왼쪽에 위치한다고 하면,
  $g(n) \in o(f(n))$.
- $c \geq 0$, $d \geq 0$, $g(n) \in O(f(n))$, 그리고 $h(n) \in \Theta(f(n))$이면, $c \times g(n) + d \times h(n) \in \Theta(f(n))$.

극한(limit)를 이용하여 차수를 구하는 방법

- **정의 :**

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \begin{cases} c > 0 \text{ 이면} & g(n) \in \Theta(f(n)) \\ 0 \text{ 이면} & g(n) \in o(f(n)) \\ \infty \text{ 이면} & f(n) \in o(g(n)) \end{cases}$$

- 예 : 다음이 성립함을 보이시오.
  - $\frac{n^2}{2} \in o(n^3)$
    $$\lim_{n\to\infty} \frac{n^2/2}{n^3} = \lim_{n\to\infty} \frac{1}{2n} = 0$$
  - $b>a>0$일 때, $a^n \in o(b^n)$
    $$\lim_{n\to\infty} \frac{a^n}{b^n} = \lim_{n\to\infty} \left(\frac{a}{b}\right)^n = 0 \quad \text{왜냐하면,} \quad 0 < \frac{a}{b} < 1$$

- 정리 : 로피탈(L'Hopital)의 법칙

$$\lim_{n\to\infty} f(n) = \lim_{n\to\infty} g(n) = \infty$$ 이면

$$\lim_{n\to\infty} \frac{g(n)}{f(n)} = \lim_{n\to\infty}\left(\frac{g'(n)}{f'(n)}\right)$$ 이다.

- 예 : 다음이 성립함을 보이시오.

$-\lg n \in o(n)$

$$\lim_{n\to\infty} \frac{\lg n}{n} = \lim_{n\to\infty}\left(\frac{\frac{1}{n\ln 2}}{1}\right) = 0$$

$-\log_a n \in \Theta(\log_b n)$

$$\lim_{n\to\infty} \frac{\log_a n}{\log_b n} = \lim_{n\to\infty}\left(\frac{\frac{1}{n\ln a}}{\frac{1}{n\ln b}}\right) = \frac{\log b}{\log a} > 0$$

2001 -03 -19 알고리즘 강의 슬라이드 1b 31