2. 알고리즘을 배우기 위한 준비

목차

2.5 알고리즘의 효율성 표현

- 알고리즘의 효율성
  - 알고리즘의 수행 시간 또는 알고리즘이 수행하는 동안 사용되는 메모리 공간의 크기로 나타낼 수 있다.
  - 이들을 각각 시간복잡도 (time complexity), 공간복잡도 (space complexity)라고 한다.
  - 일반적으로 알고리즘들을 비교할 때에는 시간복잡도가 주로 사용됨

    \+ code optimization

시간복잡도

- 시간복잡도는 알고리즘이 실행되는 동안에 사용된 기본적인 연산 횟수를 입력 크기의 함수로 나타낸다.
  - 기본 연산(Elementary Operation)이란 데이터 간 크기 비교(comparison), 데이터 읽기(read) 및 갱신(write), 숫자 계산 등과 같은 단순한 연산을 의미

- 예: 10장의 숫자 카드 중에서 최대 숫자 찾기
  - 순차탐색으로 찾는 경우에 숫자 비교가 기본적인 연산이고, 총 비교 횟수는 9번.
  - n장의 카드가 있다면, (n-1)번의 비교 수행으로 시간복잡도는 (n-1)

알고리즘의 복잡도 표현 방법

- 최악경우 분석 (Worst-case Analysis)
  - ‘어떤 입력이 주어지더라도 알고리즘의 수행시간이 얼마 이상은 넘지 않는다‘라는 상한(Upper Bound)의 의미
- 평균경우 분석 (Average-case Analysis)
  - 입력의 확률 분포를 가정하여 분석하는데, 일반적으로 균등분포(Uniform Distribution)를 가정
- 최선경우 분석 (Best-case Analysis)
  - 가장 빠른 수행시간을 분석하며, 최적(Optimal, lower bound) 알고리즘을 찾는데 활용

등교 시간 분석

- 집에서 지하철역까지 6분, 지하철을 타면 학교까지 20분, 강의실까지 걸어서 10분 걸린다.

- 최선경우
  - 집을 나와서 6분 후 지하철역에 도착하고, 운이 좋게 바로 열차를 탄 경우
  - 최선경우 시간은 6 + 20 + 10 = 36분

등교 시간 분석

- 최악경우
  - 열차에 승차하려는 순간, 열차의 문이 닫혀서 다음 열차를 기다려야 하고 다음 열차가 4분 후에 도착한다면, 최악경우는 6 + 4 + 20 + 10 = 40분

등교 시간 분석

- 위의 분석 방법이 믿을만한가?

2.6 복잡도의 점근적 표기 (Asymptotic Notation of Efficiency)

- 시간복잡도는 입력 크기에 대한 함수로 표기
  - 이 함수는 주로 여러 개의 항을 가지는 다항식
  - 이를 입력의 크기에 대한 함수로 표현하기 위해 점근적 표기 (Asymptotic Notation)를 사용

- 점근적 표기
  - 입력 크기 n이 무한대로 커질 때의 복잡도를 간단히 표현하기 위해 사용하는 표기법
  - O (Big-Oh)-표기
  - Ω (Big-Omega)-표기
  - Θ (Theta)-표기

Asymptotic Notation의 예

$$\lim_{n\to\infty}\frac{2n^2+3n}{n}$$

O(Big-Oh)-표기

- O-표기의 정의

  모든 $n \geq n_0$에 대해서 $f(n) \leq cg(n)$이 성립하는 양의 상수 c와 n_0가 존재하면, $f(n) = O(g(n))$이다.

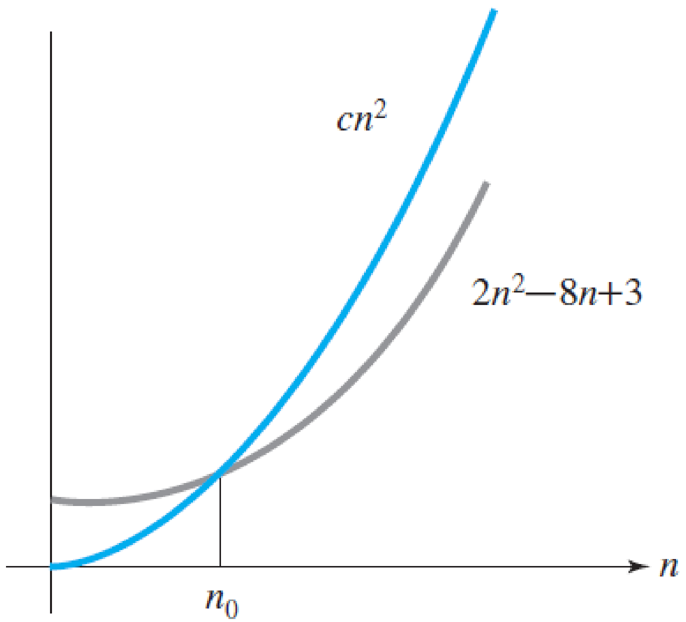
- O-표기의 의미

  n_0와 같거나 큰 모든 n (즉, n_0 이후의 모든 n) 대해서 f(n)이 cg(n)보다 크지 않다는 것

- $f(n) = O(g(n))$은 n_0 보다 큰 모든 n 대해서 f(n)이 양의 상수를 곱한 g(n)에 미치지 못한다는 뜻
- g(n)을 f(n)의 상한(Upper Bound)이라고 한다.

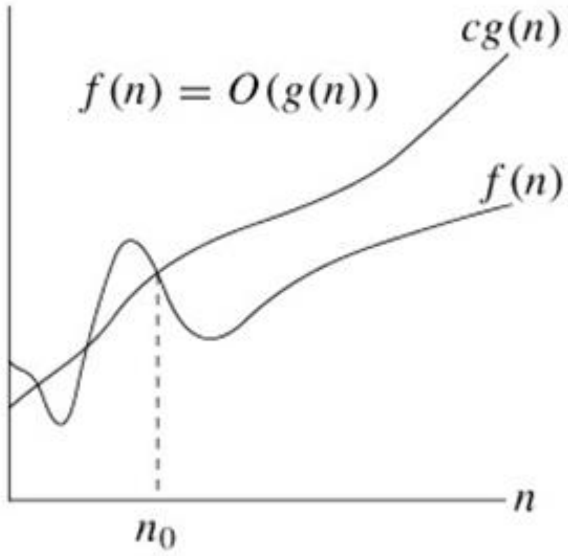
O(Big-Oh)-표기

- $f(n) = 2n^2 – 8n + 3$
  - f(n)의 단순화된 표현은 n^2
  - 단순화된 함수 n^2 에 임의의 상수 c를 곱한 c 따라 f(n)의 상한이 된다. (단, c>0.)
  - c=5일 때, $f(n) = 2n^2-8n+3$과 $5n^2$과의 교차점 (n_0=1/3)이 생기는데, 이 교차점 이후 모든 n에 대해, 즉 n이 무한대로 증가할 때, $f(n) = 2n^2-8n+3$은 $5n^2$보다 절대로 커질 수 없다.
  - 따라서 $O(n^2)$이 $2n^2-8n+3$의 점근적 상한이 된다.
  - f(n)의 O-표기는 $O(n^2)$



O(Big-Oh)-표기

- f(n) = O(g(n))
  - n이 증가함에 따라 O(g(n))이 점근적 상한이라는 것 (즉, g(n)이 n_0보다 큰 모든 n에 대해서 항상 f(n)보다 크다는 것)을 보여 준다.

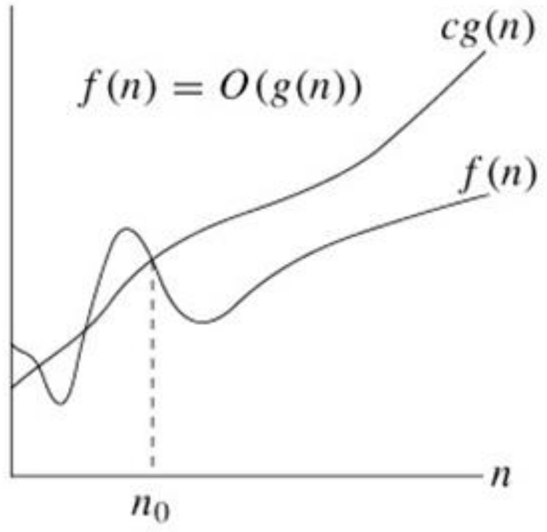


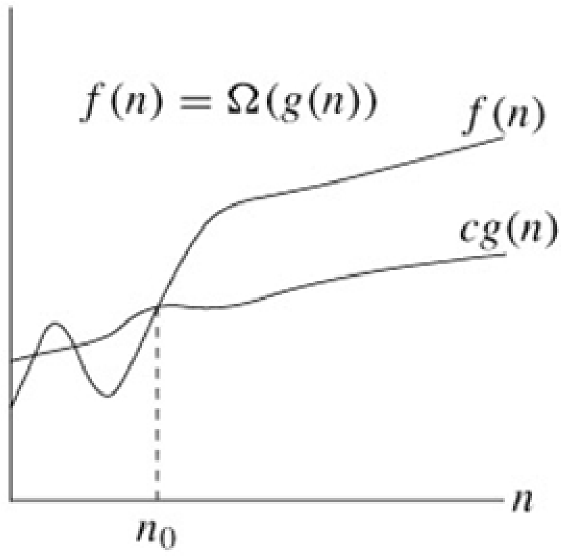
Ω(Big-Omega)-표기

- Ω-표기의 정의
  - 모든 $n \geq n_0$에 대해서 $f(n) \geq cg(n)$이 성립하는 양의 상수 c와 n_0가 존재하면, $f(n) = \Omega(g(n))$
- Ω-표기의 의미
  - n_0 보다 큰 모든 n 대해서 f(n)이 cg(n)보다 작지 않다는 것
- $f(n) = \Omega(g(n))$은 양의 상수를 곱한 g(n)이 f(n)에 미치지 못한다는 뜻
- g(n)을 f(n)의 하한(Lower Bound)이라고 한다.

Ω(Big-Omega)-표기

- $f(n) = \Omega(g(n))$
  - n이 증가함에 따라 $\Omega(g(n))$이 점근적 하한이라는 것 (즉, $g(n)$이 n_0보다 큰 모든 n에 대해서 항상 $f(n)$보다 작다는 것)을 보여준다.



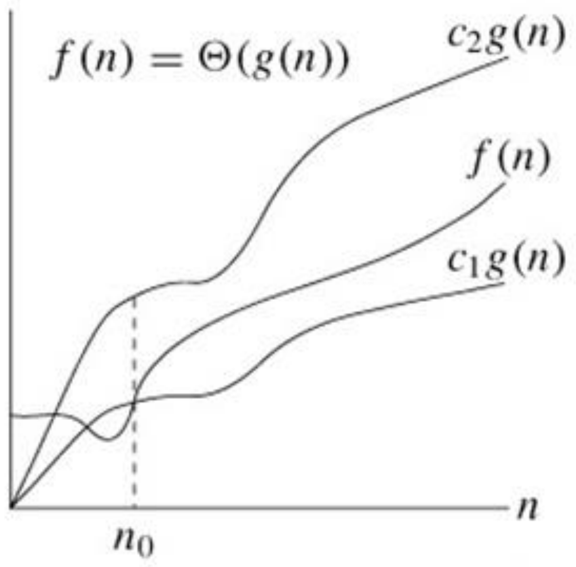


Θ(Theta)-표기

- Θ-표기의 정의
  - 모든 $n \geq n_0$에 대해서 $c_1 g(n) \geq f(n) \geq c_2 g(n)$이 성립하는 양의 상수 c_1, c_2, n_0가 존재하면, $f(n) = \Theta(g(n))$이다.
- Θ-표기의 의미
  - 수행시간의 O-표기와 Ω-표기가 동일한 경우에 사용한다. 즉 동일한 증가율을 의미
- $2n^2+3n+5=O(n^2)$과 동시에 $2n^2+3n+5=\Omega(n^2)$ 이므로, $2n^2+3n+5=\Theta(n^2)$
- $\Theta(n^2)$은 n^2과 $(2n^2+3n+5)$이 유사한 증가율을 가지고 있다는 뜻
  - 따라서 $2n^2+3n+5 \neq \Theta(n^3)$ 이고, $2n^2+3n+5 \neq \Theta(n)$이다.

Θ(Theta)-표기

- f(n)=Θ(g(n))
  - n_0보다 큰 모든 n에 대해서 Θ-표기가 상한과 하한을 동시에 만족한다는 것을 보여 준다.



Some Commonsense Rules

1. Multiplicative constants can be omitted:
   $1\cancel{4}n^2$ becomes n^2
2. n^a dominates n^b if $a > b$:

   n^2 dominates n
3. Any exponential dominates any polynomial:

   3^n dominates n^5
4. Any polynomial dominates any logarithm:
   n dominates $(\log n)^3$
   n^2 dominates $n \log n$

O(Big-Oh)-표기 예

- $n \geq 2,\ 3n+2 \leq 4n$
- $n \geq 3,\ 3n+3 \leq 4n$
- $n \geq 10,\ 100n+6 \leq 101n$
- $n \geq 5,\ 10n^2+4n+2 \leq 11n^2$
- $n \geq 4,\ 6*2^n+n^2 \leq 7*2^n$
- $n \geq 2,\ 3n+3 \leq 3n^2$
- $n \geq 2,\ 10n^2+4n+2 \leq 10n^4$

Ω(Big-Omega)-표기 예

- $n \geq 1$, $3n+2 \geq 3n$
- $n \geq 1$, $3n+3 \geq 3n$
- $n \geq 1$, $100n+6 \geq 100n$
- $n \geq 1$, $10n^2+4n+2 \geq n$
- $n \geq 1$, $6*2^n+n^2 \geq 2^n$

Θ(Theta)-표기 예

- $n \geq 2,\ 3n \leq 3n+2 \leq 4n \Rightarrow 3n+2 = \Theta(n)$
  - $c1=3,\ c2=4,\ n0=2$

- $3n+3= \Theta(n)$

- $10n^2+4n+2 = \Theta(n^2)$

- $6*2^n+n^2 = \Theta(2^n)$

자주 사용하는 O-표기

- $O(1)$ 상수 시간 (Constant time)
- $O(\log n)$ 로그(대수) 시간 (Logarithmic time)
- $O(n)$ 선형 시간 (Linear time)
- $O(n \log n)$ 로그 선형 시간 (Log-linear time)
- $O(n^2)$ 제곱 시간 (Quadratic time)
- $O(n^3)$ 세제곱 시간 (Cubic time)
- $O(2^n)$ 지수 시간 (Exponential time)

이번 시간에는...

- 알고리즘의 효율성은 주로 시간복잡도 (Time Complexity)가 사용된다.
- 시간복잡도는 알고리즘이 수행하는 기본적인 연산 횟수를 입력 크기에 대한 함수로 표현한다.
- 알고리즘의 복잡도 표현 방법:
  - 최악 경우 분석 (worst case analysis)
  - 평균 경우 분석 (average case analysis)
  - 최선 경우 분석 (best case analysis)

이번 시간에는...

- 점근적 표기 (Asymptotic Notation): 입력 크기 n이 무한대로 커질 때의 복잡도를 간단히 표현하기 위해 사용하는 표기법
- O-(Big-Oh) 표기: 점근적 상한
- Ω-(Big-Omega) 표기: 점근적 하한
- Θ-(Theta) 표기: 동일한 증가율