

3. Growth of Functions

일반적으로 알고리즘의 정확한 실행 시간을 계산할 필요가 없다. 입력이 충분히 크면 최고차항만 비교만 유의미하기 때문이다.

ex) 병합정렬($\ln n$), 삽입정렬(n^2) 처럼 이전에 공부한 애들을 보면 알 수 있음.

입력이 충분히 커서 실행 시간의 증가율의 상대적 순서만 볼때, 우리는 알고리즘의 **점근적(asymptotic)** 효율성을 따지게 된다. 입력 크기가 아주 작을 때를 제외하고는 일반적으로 최선의 선택이 될 수 있다. 즉, 일반적인 상황이란 최고차항의 차수로 실행시간의 차이를 비교하는 때를 말하는데 이 때는 점근적으로 접근하게 된다.

3.1 Asymptotic notation

알고리즘의 점근적 실행 시간을 묘사하기 위해 사용하는 표기법은 입력값이 자연수들의 집합인 함수의 관점으로 정의된다. 하지만 일반적으로 편의성 때문에 실수 범위까지 사용(abuse)하는 경우도 많다.

Asymptotic notation, functions, and running times

점근적 표기법으로 보통 알고리즘의 실행 시간을 설명한다. 삽입 정렬에서 $O(n^2)$ 가 최악의 경우 실행 시간이라고 하지만 사실 an^2+bn+c 와 같은 이차함수이자 점근적 표기법을 추상화한 것이다. 점근적 표기법은 실행 시간 뿐 아니라 사용 메모리 양 등 다양한 곳에서도 적용된다. 또한 최악의 경우 뿐 아니라 입력과 상관없이 나타낼 때도 쓰일 수 있음.

Θ -notation

챕터 2에서 봤듯, 삽입 정렬의 최악의 경우 수행시간은 입력 크기 n 에 관한 2차식이었음($O(n^2)$)

예시

함수 $g(n)$ 을 다음과 같이 정의함

$$\Theta(g(n)) = f(n) : \text{양의 상수 } c_1, c_2, n_0 \text{가 있고 } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \text{ for all } n \geq n_0.$$

Θ -표기법 : 함수를 상수 배율내에서 제한한다. $\Rightarrow c_1 g(n)$ 과 $c_2 g(n)$ 사이에 있도록!

O -표기법 : 함수의 상한을 제한 $\Rightarrow c g(n)$ 보다 밑에 있도록!

Ω -표기법 : 함수의 하한을 제한 $\Rightarrow c(g(n))$ 보다 위에 있도록!

Θ -표기법

$\Theta(g(n))$ 에서 $f(n)$ 이 사이에 있다는 것은 $f(n) \in \Theta(g(n))$ 으로 표현해야한다. $\Theta(g(n))$ 이 집합) 하지만 등호를 써서 $f(n) = \Theta(g(n))$ 이라고 표기한다. 뒤에서 이유가 나온다.

그림에서 보면 $f(n)$ 과 $c_1 g(n)$ 이 만나는 지점의 x좌표인 n_0 이후 $f(n)$ 은 항상 $\Theta(g(n))$ 사이에 있음. 이럴 때 $g(n)$ 이 $f(n)$ 의 **asymptotically tight bound** 라고 한다.

O -표기법

상한선을 제한한다는 것 = 최악의 경우 실행 시간을 제한한다는 의미.

asymptotic upper bound

Ω -표기법

asymptotic lower bound 표기

정리 1

임의의 두 함수 $f(n)$ 과 $g(n)$ 에 대해, $f(n) = \Theta(g(n))$ 일 필요충분조건은 $f(n) = O(g(n))$ 면서 $f(n) = \Omega(g(n))$ 일 때이다.

쉽게 표현하면 어떤 알고리즘의 실행 시간을 Θ 표기법으로 쓰기 위해선 실행 시간의 상한,하한 모두 동일해야 한다는 뜻!

예시 1

삽입 정렬 : 최선의 경우 실행시간은 $\Omega(n)$ 인데 이 말은 삽입 정렬의 실행 시간이 $\Omega(n)$ 임을 의미한다.

그러므로 삽입 정렬의 실행 시간은 $\Omega(n)$ 과 $O(n^2)$ 둘 다에 속하기 때문에 선형과 이차함수 사이 어딘가에 있는 값이다.

삽입 정렬의 실행시간은 $\Omega(n^2)$ 인가? => X

이미 정렬된 입력 값을 넣으면 수행 시간이 $\Theta(n)$ 이니까

단지, 어떤 입력값에서는 실행 시간이 $\Omega(n^2)$ 일 수도 있다라는 뜻.

Asymptotic notation in equations and inequalities

실행시간 표기법이 헷갈린다..

$n = O(n^2)$ 라고 쓰기도 하고 $2n^2 + 3n + 1 = 2n^2 + \Theta(n)$ 이라고도 쓴다.

점근적 표기법에서 식의 오른쪽에서 '호자' 있을 때는 위에서 말했듯 집합에 속하는 개념으로 생각한다.

$n = O(n^2)$ 는 $n \in O(n^2)$ 라는 것을 의미한다.

$2n^2 + 3n + 1 = 2n^2 + \Theta(n)$ 에서 $\Theta(n)$ 은 어떤 함수 $f(n)$ 이라고 생각하면 된다.

여기선 $f(n) = 3n + 1$ 이고, 실제로도 $\Theta(n)$ 에 속해있는 걸 알 수 있다.

2장에서 $T(n) = 2T(n/2) + \Theta(n)$ 과 같은 재귀적 표현으로 실행 시간을 적었다. 점근적 표기법의 관점에서 보면 낮은 차수들은 $\Theta(n)$ 으로 생각하면 편리하다.

Θ -표기법이 두개라면 어떨까?

$2n^2 + \Theta(n) = \Theta(n^2)$ 라는 식을 생각해보자.

위에서 $\Theta(n) = 3n + 1$ 이라고 했으니

1번식 : $2n^2 + 3n + 1 = 2n^2 + \Theta(n)$

2번식 : $\quad \quad \quad = \Theta(n^2)$

1번 식 : 위에서 처럼 $f(n) \in \Theta(n)$ 과 같은 상황을 의미

2번 식 : $g(n) \in \Theta(n)$ 이고, $2n^2 + g(n) = h(n)$ 이라면 $h(n) \in \Theta(n^2)$ 임을 의미함.

o-notation

O-notation 과 다름. 대문자 소문자.

O 에선 tight한 상한선이지만, o 에선 tight하지 않은 상한선

예시 : $2n = o(n^2)$ 는 성립하지만 $2n^2 \neq o(n^2)$ 이다.

ω -notation

O 와 o 의 관계처럼

Ω 와는 다르게 tight하지 않은 하한선 나타냄.

명제 정리

추이성

$f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n))$ 이면 $f(n) = \Theta(h(n))$ 이다.

$f(n) = O(g(n))$ and $g(n) = O(h(n))$ 이면 $f(n) = O(h(n))$ 이다.

$f(n) = \Omega(g(n))$ and $g(n) = \Omega(h(n))$ 이면 $f(n) = \Omega(h(n))$ 이다.

$f(n) = o(g(n))$ and $g(n) = o(h(n))$ 이면 $f(n) = o(h(n))$ 이다.

$f(n) = \omega(g(n))$ and $g(n) = \omega(h(n))$ 이면 $f(n) = \omega(h(n))$ 이다.

반사성

$f(n) = \Theta(f(n))$ 이다.

$f(n) = O(f(n))$ 이다.

$f(n) = \Omega(f(n))$ 이다.

대칭성

$f(n) = \Theta(g(n))$ 과 $g(n) = \Theta(f(n))$ 은 필요충분조건이다.

전치 대칭성

$f(n) = O(g(n))$ 과 $g(n) = \Omega(f(n))$ 은 필요충분조건이다.

$f(n) = o(g(n))$ 과 $g(n) = \omega(f(n))$ 은 필요충분조건이다.

점근적인 실수 비교

$f(n) = O(g(n))$ 일 때 (점근적으로) $a \leq b$ 이다.

$f(n) = \Omega(g(n))$ 일 때 (점근적으로) $a \geq b$ 이다.

$f(n) = \Theta(g(n))$ 일 때 (점근적으로) $a = b$ 이다.

$f(n) = o(g(n))$ 일 때 (점근적으로) $a < b$ 이다.

$f(n) = \omega(g(n))$ 일 때 (점근적으로) $a > b$ 이다.

Exercise

문제 3.1-1

$f(n)$ 과 $g(n)$ 이 점근적으로 음수가 아닌 함수라고 하자. Θ -표기법의 기본 정의를 이용하여

$\max(f(n), g(n)) = \Theta(f(n) + g(n))$ 임을 증명하라.

답

어떤 상수 c 와 C 에서 다음이 성립함을 증명하면 된다.

$$c(f(n)+g(n)) \leq \max(f(n), g(n))$$

$$\max(f(n), g(n)) \leq C(f(n) + g(n))$$

첫번째의 경우 c 가 $1/2$ 라면 성립함을 알 수 있다.

$f(n)$ 과 $g(n)$ 중 작은 것이 큰 것의 절반을 채워줄 수 없기 때문이다.

두번째 식은 두 값이 음이 아닌 실 수 이므로 항상 성립함을 알 수 있다.

문제 3.1-2

b 가 양수라면 어떤 실수 a와 b에서든 다음이 성립함을 보여라

$$(n+a)^b = \Theta(n^b)$$

답

$c(n+a)^b \leq n^b \leq C(n+a)^b$ 가 성립해야한다.

n이 a의 영향을 받지 않을 정도로 충분히 크다면 c와 C를 $0 \leq c \leq 1, 1 \leq C$ 로 두면 성립한다.

문제 3.1-3

왜 명제 " 알고리즘 A의 실행 시간은 최소한 $O(n^2)$ 이다."가 무의미한지 설명하라

답

O -표기법은 점근적 표기법에서 상한선을 의미한다.

즉 최소한이 아니라 최대라고 해야 맞는 말이기때문에 위 명제는 잘못되어 의미가 없다.

문제 3.1-4

$2^{n+1} = O(2^n)$ 이 참인가? $2^{2n} = O(2^n)$ 은 참인가?

답

n이 충분히 큰 입력 크기라면 첫번째 문장은 계수가 2인 것 뿐이기때문에 참이라고 할 수 있다.

두번째 문장은 2^{2n} 을 제곱한 형태기때문에 다르다. 거짓

문제 3.1-5

정리 1 : 임의의 두 함수 f(n)과 g(n)에 대해, $f(n) = O(g(n))$ 일 필요충분조건은 $f(n) = O(g(n))$ 면서 $f(n) = \Omega(g(n))$ 일 때이다.

을 증명하라.

답

$f(n) = O(g(n))$ 은 f(n)의 상한이 g(n)이라는 의미이고

$f(n) = \Omega(g(n))$ 은 f(n)의 하한이 g(n)이지만 tight하지 않음을 의미한다.

Θ -표기법에서 등호의 의미는 포함된다는 의미이므로 두 문장은 f(n)의 상한과 하한을 말하는 명제로 서로 필요충분조건이다.

문제 3.1-6

알고리즘의 실행 시간이 $\Theta(g(n))$ 인 것과 최악의 경우 수행시간이 $O(g(n))$ 이고 최선의 경우 수행시간이 $\Omega(g(n))$ 인 것이 필요충분조건임을 증명하라.

답

$\Theta(g(n))$ 의 의미는 상한과 하한을 정해준 것이다.

O -표기법은 상한을 정한다. 즉 최악의 경우 가장 수행시간이 높고 growth도 급격할때의 한계선이고

Ω -표기법은 하한을 정한다. 즉 최선의 경우 가장 수행시간이 낮고 growth도 완만할때의 한계선이다.

따라서 둘은 서로 필요충분조건이다.

문제 3.1-7

$\alpha(g(n))$ 과 $\omega(g(n))$ 의 교집합이 공집합임을 증명하라.

답

$\alpha(g(n))$ 은 $O(g(n))$ 의 tight하지 않은 버전이라고 할 수 있고

$\omega(g(n))$ 은 $\Omega(g(n))$ 의 tight하지 않은 버전이라고 할 수 있다.

정확한 의미를 말하자면 첫번째 예선 $f(n)$ 이 $g(n)$ 보다 같지 않고 확실히 더 작은 growth를 보이고

두번째예선 $f(n)$ 이 $g(n)$ 보다 확실히 같지 않고 더 큰 growth를 보인다는 의미이다.

둘은 공존할 수 없으므로 교집합은 공집합이다.

문제 3.1-8

우리는 우리의 표기법을 각자 다른 기호기로 무한대로 수렴할 수 있는 두 인자 n 과 m 의 경우까지 확대시킬 수 있다. 주어진 함수 $g(n,m)$ 에 대해 $O(g(n,m))$ 을 다음과 같이 정의한다.

$O(g(n,m)) = \{ f(n,m) : \text{양수인 상수 } c, n_0, m_0 \text{이 있을 때}$

모든 $n \geq n_0$ 혹은 $m \geq m_0$ 에 대하여

$0 \leq f(n,m) \leq cg(n,m)$ 이 성립한다.}

$\Omega(g(n,m))$ 과 $\Theta(g(n,m))$ 에 대해 정의하라.

답

O -표기법은 상한을 나타낸다.

Ω 는 하한을 나타내고 Θ 는 범위를 나타내므로

$\Omega(g(n,m)) = f(n,m) : \text{양의 상수 } c_1, n_0, m_0 \text{가있고 } 0 \leq cg(n,m) \leq f(n,n) \text{ 가 } m \geq m_0 \text{ 그리고 } n \geq n_0 \text{ 에서성립한다..}$

$\Theta(g(n,m)) = f(n,m) : \text{양의 상수 } c_1, c_2, n_0, m_0 \text{가있고 } 0 \leq c_1g(n,m) \leq f(n,m) \leq c_2g(n,m) \text{ 가 } m \geq m_0 \text{과 } n \geq n_0 \text{에서 성립한다}$