2022110151 이준연 알고리즘 3분반 과제1

Problem 0

Problem 0

$f_1(n) = O(g_1(n)),\quad f_2(n) = O(g_2(n))$

(1) $f_1(n) + f_2(n) = O(g_1(n) + g_2(n)) = O(\max(g_1(n), g_2(n)))$

(2) $f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$ 증명

- Prove statement (2).

∨ 정의에 따라 $f_1(n) = O(g_1(n))$은 (어떤 양의 상수 c_1과 n_1 존재하므로)
모든 $n \geq n_1$에 대해 $f_1(n) \leq c_1 \cdot g_1(n)$ 이다

∨ $f_2(n) = O(g_2(n))$도 또한 어떤 양의 상수 c_2와 n_2가 존재하므로)
모든 $n \geq n_2$에 대해 $f_2(n) \leq c_2 \cdot g_2(n)$이다

→ (두 식을 곱하면)

$f_1(n) \times f_2(n) \leq (c_1 \cdot g_1(n)) \times (c_2 \cdot g_2(n))$
$= (c_1 \cdot c_2) \cdot g_1(n) \cdot g_2(n)$ 이고,

이 부등식은 $n \geq \max(n_1, n_2)$ 일 때 항상 성립하므로 (상수 c_1, c_2는 n에 영향 X)

∴ $f_1(n) \cdot f_2(n) = O(g_1(n) \cdot g_2(n))$ 이라고 표현할 수 있다.

Problem 1

Problem 1

- Show that $5n^2 = O(n^2)$. 빅오 Big-O 정의 이용 증명하기

모든 $n \geq n_0$에 대해 $5n^2 \leq c \cdot n^2$라고 표현할 수 있다.
(0이 아닌) n에 대해 양변을 n^2으로 나누면 → $5 \leq c$ 이고,
이 부등식을 만족하는 상수 c를 6으로, n_0은 1로 고정할 때 모든 $n \geq 1$에 대해
→ $5n^2 \leq 6n^2$ 이므로

∴ $5n^2 = O(n^2)$이라고 말할 수 있다.

(참고로 n은 보통 문제/입력의 크기를 나타내는 변수이므로 $n \geq 1$과 같이 $n \geq n_0$을 바탕으로 설명하였다)
($n \geq 0$ 이 아닌)

Problem 2

Problem 2

- Show that $2n^2 - 10n + 3 = \Theta(n^2)$.

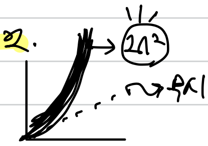
Hint) You can use the proofs of Big-O (O) and Big-Omega (Ω).   상한 (Big-O)   하한 (Big-Omega)

* 빅세타 정의 ($\underbrace{f(n) = O(g(n))}_{①}$ & $\underbrace{f(n) = \Omega(g(n))}_{②}$ 활용, 보이기,,

⓪ $f(n) = 2n^2 - 10n + 3$

V ① 상한 : Big-O 만족함을 보이기

$\Rightarrow f(n) = 2n^2 - 10n + 3 \le c \cdot n^2$ ⇒ 만족하는 상수 c와 n_0 찾을 것.

ex) 

$\Rightarrow$ 충분히 큰 n에 대해 $2n^2 - 10n + 3 \le 2n^2$

(n이 충분히 큰 경우, $-10n+3$은 $2n^2$에 비해 묻혀진다 (무시된다)).

(이유) e.g. $c = 3$, $n \ge 10$ 이라 가정하면,

$2n^2 - 10n + 3 \le 3n^2$ 이므로

$\therefore f(n) = O(n^2)$이라고 할 수 있다.

V ② 하한 : Big-Omega 만족함 보이기

$\Rightarrow 2n^2 - 10n + 3 \ge c \cdot n^2$ ⇒ 만족하는 상수 c와 n_0을 찾을 것!

$\Rightarrow$ 충분히 큰 n에 대해 $2n^2 - 10n + 3 \ge n^2$

(이유) $(2n^2 - 10n + 3) - n^2 \ge \underbrace{\cancel{n^2} - \cancel{n^2}}_{0}$

$\Rightarrow n^2 - 10n + 3 \ge 0$ 이면 되는데,

e.g $n \ge 11$ 인 경우, $n^2 - 10n + 3 \ge 0$ 이므로

$121 - 110 + 3$

$\therefore f(n) \ge n^2 \Rightarrow f(n) = \Omega(n^2)$ 이라고 말할 수 있다!

$\therefore$ ①과 ② 모두 만족하므로 $f(n) = 2n^2 - 10n + 3 = O(n^2)$ 이라고 결론할 수 있다.

Big-O Big-Omega

Problem 3

Problem 3

a. O(n) b. Ω(n) c. θ(n) d. O(n^2) e. Ω(n^2) f. θ(n^2)

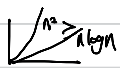
- For each of the following, select all related items from the options above.

1. 8n
2. 8n-3
3. n^2+3nlogn
4. 4nlogn
5. 5n^2+3
6. n^3+3nlogn
7. n^{2}logn +n

1. $8n$: a. $O(n)$ b. $\Omega(n)$ c. $\Theta(n)$ — 이유: 정확히 n에 비례하기 때문

2. $8n-3$: a. $O(n)$ b. $\Omega(n)$ c. $\Theta(n)$ — 이유: n이 커질수록 -3은 무시가능하므로 결국 $\Theta(n)$에 해당된다.

3. $n^2+3n\log n$: d. $O(n^2)$ e. $\Omega(n^2)$ f. $\Theta(n^2)$



비교적

이유 : n이 커질수록 nlogn은 무시되므로 $n^2 \geq n\log n$
n^2 고정

4. $4n\log n$: a. $O(n)$ d. $O(n^2)$, b. $\Omega(n)$ — 이유: $n\log n < n^2$ 이므로 $\Theta(n\log n)$이지만 $O(n^2)$도 된다..

5. $5n^2+3$: d. $O(n^2)$ e. $\Omega(n^2)$ f. $\Theta(n^2)$ — 이유: 최고차항이 n^2이므로 3은 무시된다.

6. $n^3+3n\log n$: 없음 — 이유: 최고차항이 n^3이고 나머지는 무시되므로 $O(n^3)$, $\Omega(n^3)$, $\Theta(n^3)$이지만 n^3은 보기에 선택지가 없다.

7. $n^2\log n+n$: d. $O(n^2)$ (느낌에 보았을때 가능하다...) — 이유: $n^2\log n \geq n$
$\Theta(n^2\log n)$이고 $O(n^3)$ 가능하지만 보기에 없다...
만약 보기중 골라야 한다면 d. $O(n^2)$를 고를수 있을 것 같다...

Problem 4

Problem 4

- What function does the running time of the following algorithm correspond to when the input size is n? (Express the answer using Θ notation)

```
sample(A[ ], n)
{
    sum1 ← 0 ;
①   for i ← 1 to n
        sum1← sum1+ A[i];
    sum2 ← 0 ;
②-1 for i ← 1 to n
②-2     for j ← 1 to n
            sum2← sum2+ A[i]*A[j] ;

    return sum1+sum2 ;
}
```

①번 for 루프 : 반복횟수 n번 & 각반복은 상수시간
⇒ ∴ 시간복잡도 : $\Theta(n)$ → $\Theta(n)$

②-1번 for 루프 (바깥) : 반복횟수 n번
②-2번 for 루프 (안쪽) : 각 바깥루프마다 n번
⇒ ②번 총반복횟수 : $n \times n = n^2$
(각반복은 상수시간)
⇒ ∴ 시간복잡도 : $\Theta(n^2)$ → $\Theta(n^2)$

∴ $\Theta(n+n^2) = \Theta(n^2)$
(n^2이 n보다 빠르게 커지므로, n은 무시하고 최고차항 n^2만 고려)

$\boxed{\therefore \Theta(n^2)}$