COSE213: Data Structure

Lecture 2 Review

Minseok Jeon

2024 Fall

10/11 수업

- 10/11 수업은 12시 50분에 시작합니다.

Questions

Definition. *For two functions f(n) and g(n), we say that* $f(n) = O(g(n))$ *or* $f(n) \in O(g(n))$ *if there exist positive constant* c *and* n_0 *such that*

$$\forall n \geq n_0 \,.\, 0 \leq f(n) \leq c * g(n)$$

- 질문: 아래 명제는 왜 valid 한가요?
  - $T(n) \in O(n) \implies T(n) \in O(n^2)$

proof.

- $T(n) \in O(n)$이 참이라는 것은 아래 식을 만족하는 n_0와 c가 존재한다는 것이다.

$$\forall n \geq n_0 \,.\, 0 \leq T(n) \leq c * n \qquad (1)$$

- 위 식을 만족하는 n_0와 c를 각각 a와 b라고 하자. 즉 아래 식이 성립한다.

$$\forall n \geq a \,.\, 0 \leq T(n) \leq b * n \qquad (2)$$

Questions

- $(a + 1) > a$이기 때문에 위 식이 성립한다는 것은 아래의 식도 성립한다는 것을 의미한다.

$$\forall n \geq (a+1) \,.\, 0 \leq T(n) \leq b * n \qquad (3)$$

- 또한, 1 이상의 모든 실수에 대해서 아래의 식이 성립한다.

$$\forall n \geq 1 \,.\, b * n \leq b * n^2 \qquad (4)$$

- 식 (3)과 (4)가 성립하기 때문에 아래의 식이 성립한다.

$$\forall n \geq (a+1) \,.\, 0 \leq T(n) \leq b * n \leq b * n^2 \qquad (5)$$

- 식 (5)가 성립하기 때문에 아래의 식이 성립한다. 즉 $T(n) \in O(n^2)$이기 위한 n_0와 c가$a + 1, b$로 존재한다.

$$\forall n \geq (a+1) \,.\, 0 \leq T(n) \leq b * n^2 \qquad (6)$$

- 따라서, $T(n) \in O(n) \implies T(n) \in O(n^2)$ 이다.

Questions

- 질문: 아래 명제는 valid 한가요?

$$(T(n) \in O(n) \wedge T(n) \in \Omega(n)) \implies T(n) \in \Theta(n)$$

proof.

- $T(n) \in O(n)$이 참이라는 것은 아래 식을 만족하는 n_0와 c가 존재한다는 것이다.

$$\forall n \geq n_0 \,.\, 0 \leq T(n) \leq c * n \qquad (1)$$

- 위 식을 만족하는 n_0와 c를 각각 a와 b라고 하자. 즉 아래 식이 성립한다.

$$\forall n \geq a \,.\, 0 \leq T(n) \leq b * n \qquad (2)$$

- $T(n) \in \Omega(n)$이 참이라는 것은 아래 식을 만족하는 n_0'와 c'가 존재한다는 것이다.

$$\forall n \geq n_0' \,.\, T(n) \geq c' * n \qquad (3)$$

- 위 식을 만족하는 n_0'와 c'를 각각 a'와 b'라고 하자. 즉 아래 식이 성립한다.

$$\forall n \geq a' \,.\, T(n) \geq b' * n \qquad (4)$$

Questions

- a와 a'중 더 큰 수를 a''이라고 하자 그럼 식 (2)와 (4)에 의해 아래의 식이 성립한다.

$$\forall n \geq a'' .\, b' * n \leq T(n) \leq b * n \qquad (5)$$

- 즉 아래 식을 만족하게 하는 n_0, c_1, c_2가 a'', b', b로 존재한다.

$$\forall n \geq n_0 .\;\; c_1 * g(n) \leq T(n) \leq c_2 * g(n) \qquad (6)$$

- 따라서, $(T(n) \in O(n) \wedge T(n) \in \Omega(n)) \implies T(n) \in \Theta(n)$ 이다.

Questions

- 질문: 그렇다면 아래 명제는 valid 한가요?

$$(T(n) \in O(n) \wedge T(n) \in \Omega(1)) \implies T(n) \notin \Theta(n)$$

disproof.

- $T(n) = n$이라고 하자 그럼 아래 식을 만족하는 n_0와 c가 1,1로 존재하므로 $T(n) \in O(n)$이다.

$$\forall n \geq n_0 \,.\, 0 \leq n \leq c * n$$

- 또한 아래 식을 만족하는 n_0와 c가 1,1로 존재하므로 $T(n) \in \Omega(1)$이다.

$$\forall n \geq n_0 \,.\, n \geq c * 1$$

- $T(n) = n$일 경우 아래의 식을 만족하는 n_0, c_1, c_2가 1,1,1로써 존재한다. 즉 반례가 있다.

$$\forall n \geq n_0 \,.\; c_1 * n \leq n \leq c_2 * n$$

Questions

- 질문: 아래 알고리즘의 복잡도는 왜 $O(n)$인가요?

```
procedure search(⟨d_1,d_2,…,d_n⟩, val)
  idx ← 0
  for i ←1 to n do
    if d_i = val then
      return i
    end if
  return -1
end procedure
```

Questions

- 질문: 아래 알고리즘에서 d_{idx}는 무슨 뜻인가요?

procedure read($\langle$d$_1$,d$_2$,...,d$_n\rangle$, idx)
  return d$_{idx}$
end procedure

Examples

- 아래 명제가 valid함을 보이시오.
    - $T(n) \in O(n) \implies T(n) \in O(2^n)$

Examples

- 아래 명제가 valid함을 보이시오.
  - $T(n) \in \Omega(n^2) \implies T(n) \in \Omega(n)$

Examples

- 아래 명제가 valid하지 않음을 보이시오.
    - $(T(n) \in O(n^2) \wedge T(n) \in \Omega(n)) \implies T(n) \notin \Theta(n^2)$