# `[SWE2015-41]` Asymptotic Notations

Department of Computer Science and Engineering, Sungkyunkwan University

Spring 2025

## 1 Definitions of Asymptotic Notations

**Definition 1.** $f(n) = O(g(n))$ if $\exists c > 0, \exists n_0 \in \mathbb{N}$ such that $f(n) \leq cg(n)$ for all $n \geq n_0$.

**Definition 2.** $f(n) = \Omega(g(n))$ if $\exists c > 0, \exists n_0 \in \mathbb{N}$ such that $f(n) \geq cg(n)$ for all $n \geq n_0$.

**Definition 3.** $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

**Note.** Formally, $f(n) \in O(g(n))$ makes more sense than $f(n) = O(g(n))$ because $O(g(n))$ is *a set of functions* increasing not faster than $g(n)$. However, in this course, it is okay to use the latter.

## 2 Examples

**Proposition 1.** $n^2 + 10n = O(n^2)$.
*Proof.* To prove the above statement, we need to find some $c > 0$ and $n_0 \in \mathbb{N}$ satisfying:

$$\underbrace{n^2 + 10n}_{f(x)} \leq c \times \underbrace{n^2}_{g(x)} \quad \text{for all} \quad n \geq n_0.$$

Choose $c = 2$ and $n_0 = 10$. Then, for all $n \geq n_0$,

$$cg(n) = c \cdot n^2 = 2n^2 = n^2 + n^2 \geq n^2 + 10n = f(n).$$

Therefore, by Definition 1, $n^2 + 10n = O(n^2)$. □

**Proposition 2.** $n^3 + n + 5 \neq O(n^2)$.
*Proof.* Given any $c$ and $n_0$, if we set $n = \lceil \max(c, n_0) \rceil + 1$, the following statements are always true:

$$\underbrace{n^3 + n + 5}_{f(n)} > n^3 > c \times \underbrace{n^2}_{g(N)} \quad \text{and} \quad n > n_0.$$

This proves that for any $c > 0$ and $n_0 \in \mathbb{N}$, there exists some $n > n_0$ such that $f(x) > cg(n)$. Namely, $n^3 + n + 5 \neq O(n^2)$. □

**Proposition 3.** $n = O(2^n)$.
*Proof.* We can prove $2^n \geq n$ for all $n \geq 1$ using mathematical induction (수학적 귀납법).
(Base case) $2^n \geq n$ trivially holds when $n = 1$.
(Induction step) Assume $2^n \geq n$ holds for some $n \geq 1$. Then, $2^{(n+1)} = 2 \times 2^n \geq 2n = n + n \geq n + 1$.
By mathematical induction, $2^n \geq n$ holds for all $n \geq 1$. By definition, $n = O(2^n)$. □

**Proposition 4.** $f(n) = O(h(n))$ if $f(n) = O(g(n))$ and $g(n) = O(h(n))$.
*Proof.* By Definition 1,

$$\exists c_1 > 0,\ \exists n_1 \in \mathbb{N} \quad \text{such that} \quad \forall n \geq n_1,\ f(n) \leq c_1 g(n),$$
$$\exists c_2 > 0,\ \exists n_2 \in \mathbb{N} \quad \text{such that} \quad \forall n \geq n_1,\ g(n) \leq c_2 h(n).$$

Now, choose $c = c_1 \times c_2$ and $n_0 = \max(n_1, n_2)$. Then, for all $n \geq n_0$, one can derive the following inequality: $f(n) \leq c_1 g(n) \leq c_1 c_2 h(n) = ch(n)$. By definition again, $f(n) = O(h(n))$. □