CS G526: Advanced Algorithms and Complexity

Lecture 2
Asymptotic Notations and Graphs

Instructor: Prof. Aniket Basu Roy
Scribes: Amoghavarsha and Sakshi

August 2026

1 Asymptotic Notations

Asymptotic notation provides a mathematical way to describe the growth of functions as the input size n becomes large.

In algorithm analysis, we are interested in how the running time or space requirement of an algorithm grows with respect to the input size. Asymptotic notation allows us to compare growth rates while ignoring constant factors and lower-order terms.

The three fundamental asymptotic notations are

$$O(\cdot), \qquad \Omega(\cdot), \qquad \Theta(\cdot).$$

We also consider the stricter relations

$$o(\cdot), \qquad \omega(\cdot).$$

1.1 Big-O Notation

Definition

We say that

$$f(n) = O(g(n))$$

if there exist constants $c > 0$ and $n_0 > 0$ such that

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

for every $n \geq n_0$.
Formally,

$$\boxed{\exists c > 0,\ \exists n_0 > 0,\ \forall n \geq n_0 : f(n) \leq c\, g(n)}$$

Big-O gives an **asymptotic upper bound**. The condition only needs to hold for sufficiently large n.

For example,

$$n = O(n^2)$$

because for $n \geq 1$,

$$n \leq n^2.$$

The constants c and n_0 are not unique.

1.2 Graphical Interpretation of Big-O

If

$$f(n) = O(g(n)),$$

then there exist $c > 0$ and $n_0 > 0$ such that

$$f(n) \le c\,g(n) \qquad \forall n \ge n_0.$$

Thus, after n_0, the graph of $f(n)$ remains below $c\,g(n)$.

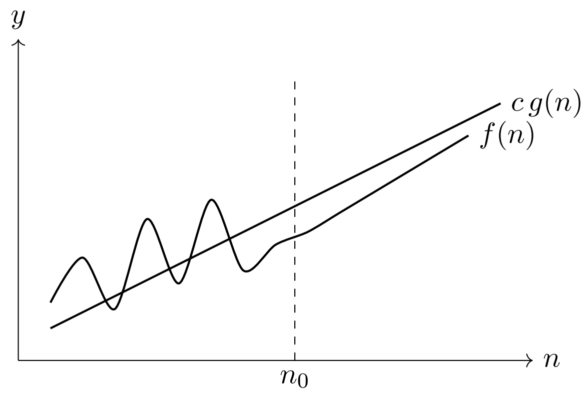

Figure 1: Graphical interpretation of $f(n) = O(g(n))$. The functions may cross before n_0; for $n \ge n_0$, $f(n) \le c\,g(n)$.

The functions may cross before n_0; only their behavior for $n \ge n_0$ matters.

1.3 Big-Omega Notation

Definition

We say that

$$f(n) = \Omega(g(n))$$

if there exist constants $c > 0$ and $n_0 > 0$ such that

$$f(n) \ge c\,g(n)$$

for every $n \ge n_0$.
Formally,

$$\boxed{\exists c > 0,\ \exists n_0 > 0,\ \forall n \ge n_0 : f(n) \ge c\,g(n)}$$

Big-Omega gives an **asymptotic lower bound**.

For example,

$$n^2 = \Omega(n)$$

because

$$n^2 \geq n \qquad \forall n \geq 1.$$

1.4 Graphical Interpretation of Big-Omega

If

$$f(n) = \Omega(g(n)),$$

then after n_0,

$$f(n) \geq c\,g(n).$$

Thus, the graph of $f(n)$ remains above $c\,g(n)$ for all sufficiently large n.

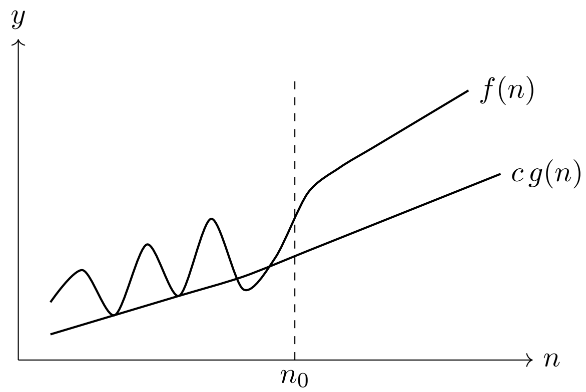

Figure 2: Graphical interpretation of $f(n) = \Omega(g(n))$. The functions may cross before n_0; for $n \geq n_0$, $f(n) \geq c\,g(n)$.

1.5 Big-Theta Notation

Definition

We say that

$$f(n) = \Theta(g(n))$$

if there exist constants $c_1, c_2 > 0$ and $n_0 > 0$ such that

$$c_1 g(n) \leq f(n) \leq c_2 g(n)$$

for every $n \geq n_0$.
Formally,

$$\exists c_1, c_2 > 0,\ \exists n_0 > 0,\ \forall n \geq n_0 : \quad c_1 g(n) \leq f(n) \leq c_2 g(n)$$

Big-Theta gives a **tight asymptotic bound**. It means that $f(n)$ and $g(n)$ have the same asymptotic growth rate up to constant factors.

The important relationship is

$$\boxed{f(n) = \Theta(g(n)) \iff f(n) = O(g(n)) \text{ and } f(n) = \Omega(g(n)).}$$

1.6 Graphical Interpretation of Big-Theta

If

$$f(n) = \Theta(g(n)),$$

then after n_0,

$$c_1 g(n) \leq f(n) \leq c_2 g(n).$$

Thus, $f(n)$ is trapped between two constant multiples of $g(n)$.

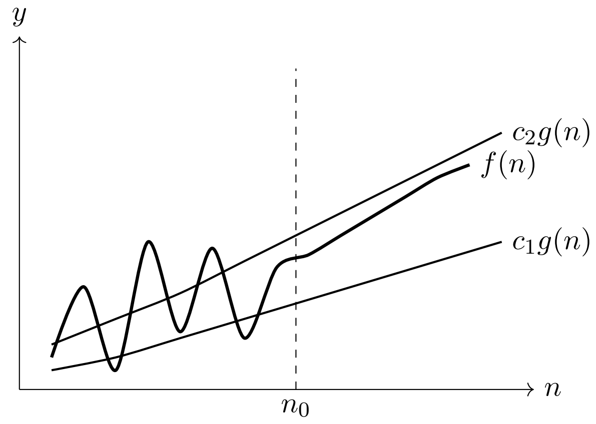

Figure 3: Graphical interpretation of $f(n) = \Theta(g(n))$. Before n_0, $f(n)$ may cross either bounding function; for $n \geq n_0$, it remains between $c_1 g(n)$ and $c_2 g(n)$.

1.7 Comparison of Asymptotic Relations

Notation	**Meaning**
$O(g(n))$	Asymptotic upper bound
$\Omega(g(n))$	Asymptotic lower bound
$\Theta(g(n))$	Tight asymptotic bound

The conditions need only hold for sufficiently large n.

1.8 Examples

Consider

$$n = O(n^2).$$

This is true since $n \leq n^2$ for $n \geq 1$.

However,

$$n \neq \Omega(n^2),$$

because n grows more slowly than n^2.

Also,

$$10^6 = \Theta(1)$$

and

$$10^6 n = \Theta(n).$$

Therefore, constant factors do not change asymptotic growth.

1.9 Transitivity

Asymptotic relations satisfy useful transitivity properties.

If

$$f(n) = O(g(n))$$

and

$$g(n) = O(h(n)),$$

then

$$\boxed{f(n) = O(h(n)).}$$

Similarly,

$$f(n) = \Theta(g(n)) \quad \text{and} \quad g(n) = \Theta(h(n))$$

imply

$$\boxed{f(n) = \Theta(h(n)).}$$

The same transitivity property also holds for Ω.

1.10 Asymptotic Equivalence

The relation Θ behaves like an equivalence relation because it is:

- reflexive,
- symmetric, and

- transitive.

For example,

$$6n^2, \qquad 10n^2, \qquad 9n^2$$

all have the same asymptotic growth rate:

$$6n^2 = \Theta(n^2),$$

$$10n^2 = \Theta(n^2),$$

$$9n^2 = \Theta(n^2).$$

Hence,

$$6n^2 = \Theta(10n^2).$$

1.11 Little-o Notation

Little-o is a stricter form of asymptotic upper bound.

Definition

We say

$$f(n) = o(g(n))$$

if, for every constant $c > 0$, there exists $n_0 > 0$ such that

$$f(n) < c\, g(n)$$

for every $n \geq n_0$.

Equivalently, when the ratio is defined,

$$\boxed{f(n) = o(g(n)) \iff \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0.}$$

For example,

$$\frac{n}{n^2} = \frac{1}{n} \to 0.$$

Therefore,

$$\boxed{n = o(n^2).}$$

Notice that

$$n = O(10^6 n)$$

but

$$n \neq o(10^6 n),$$

because

$$\lim_{n\to\infty} \frac{n}{10^6 n} = \frac{1}{10^6} \neq 0.$$

1.12 Little-Omega Notation

Little-Omega is the strict counterpart of Big-Omega.

Definition

We say

$$f(n) = \omega(g(n))$$

if, for every constant $c > 0$, there exists $n_0 > 0$ such that

$$f(n) > c\, g(n)$$

for every $n \geq n_0$.

Equivalently,

$$\boxed{f(n) = \omega(g(n)) \iff \lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty.}$$

For example,

$$\frac{n^2}{n} = n \to \infty,$$

so

$$\boxed{n^2 = \omega(n).}$$

1.13 Summary of Asymptotic Notations

Notation	Meaning	Limit intuition
$O(g(n))$	Upper bound	Ratio bounded above
$\Omega(g(n))$	Lower bound	Ratio bounded below
$\Theta(g(n))$	Tight bound	Same growth rate
$o(g(n))$	Strictly smaller	$f/g \to 0$
$\omega(g(n))$	Strictly larger	$f/g \to \infty$

The key distinction is:

$$O, \Omega, \Theta$$

are non-strict asymptotic relations, whereas

$$o, \omega$$

represent strict asymptotic growth relationships.

2 Graph and Tree Basics

A graph is a mathematical structure used to represent relationships between objects. It is written as

$$G = (V, E),$$

where V is the set of vertices and E is the set of edges.

2.1 Graph Definition

Definition

A graph is a pair

$$G = (V, E),$$

where

- V is a set of vertices (nodes), and
- E is a set of edges representing relationships between vertices.

2.2 Directed and Undirected Graphs

In a **directed graph**, every edge has a direction.

An edge from v_i to v_j is represented by

$$(v_i, v_j).$$

In general,

$$(v_i, v_j) \neq (v_j, v_i).$$

The first vertex is the source and the second is the destination.

In an **undirected graph**, an edge has no direction and is represented by

$$\{v_i, v_j\}.$$

Thus,

$$\{v_i, v_j\} = \{v_j, v_i\}.$$

Hence,

$$\boxed{\text{Directed: order matters}}$$

whereas

$$\boxed{\text{Undirected: order does not matter.}}$$

2.3 Self-Loops, Multiple Edges and Simple Graphs

A **self-loop** is an edge connecting a vertex to itself:

$$(v_i, v_i).$$

Multiple edges occur when more than one edge connects the same pair of vertices.

A **simple graph** is an undirected graph containing

- no self-loops, and
- no multiple edges.

2.4 Subgraphs

Let

$$G = (V, E).$$

A graph

$$H = (V', E')$$

is a subgraph of G if

$$V' \subseteq V \qquad \text{and} \qquad E' \subseteq E.$$

Thus, a subgraph can be obtained by selecting some vertices and some edges from the original graph.

2.5 Induced Subgraphs

Let

$$V' \subseteq V.$$

The induced subgraph on V' contains

- exactly the vertices in V', and
- every edge of G whose two endpoints belong to V'.

For an undirected graph,

$$E' = \{\{u, v\} \in E \mid u, v \in V'\}.$$

Thus, once the vertices are selected, all edges between those vertices that existed in the original graph must also be included.

2.6 Paths

A path is a sequence

$$v_0, v_1, \ldots, v_k$$

such that every consecutive pair is connected by an edge.

For an undirected graph,

$$\{v_{i-1}, v_i\} \in E.$$

For a directed graph,

$$(v_{i-1}, v_i) \in E.$$

In the context of this lecture, a path does not repeat vertices.

For example,

$$v_1, v_2, v_3, v_4$$

is a path if

$$\{v_1, v_2\},\ \{v_2, v_3\},\ \{v_3, v_4\}$$

are edges.

2.7 Connectivity

An undirected graph

$$G = (V, E)$$

is **connected** if for every pair of vertices

$$u, v \in V,$$

there exists a path from u to v.

Formally,

$$\boxed{\forall u, v \in V, \quad \exists \text{ a path from } u \text{ to } v.}$$

If such a path does not exist for some pair, the graph is disconnected.

2.8 Strong Connectivity in Directed Graphs

A directed graph is **strongly connected** if for every pair of vertices u, v, there exists a directed path from u to v.

Thus,

$$\boxed{\forall u, v \in V, \quad \exists \text{ a directed path from } u \text{ to } v.}$$

The direction of the edges must be respected.

2.9 Cycles

A **cycle** is a closed path that starts and ends at the same vertex, with no other vertex repeated.

For example,

$$v_1, v_2, v_3, v_1$$

forms a cycle if

$$\{v_1, v_2\},\ \{v_2, v_3\},\ \{v_3, v_1\} \in E.$$

A graph is called **acyclic** if it contains no cycle.

2.10 Trees

Definition

A **tree** is a connected undirected graph containing no cycles.
Equivalently,

$$\boxed{\text{Tree} \iff \text{Connected and Acyclic.}}$$

A tree has the following important properties:

- If a tree has n vertices, it has exactly $n-1$ edges.
- There is exactly one simple path between every pair of vertices.
- Adding an edge between any two existing vertices creates a cycle.
- Removing any edge disconnects the tree.

Therefore,

$$\boxed{|E| = |V| - 1.}$$

For example, a tree with 5 vertices has

$$5 - 1 = 4$$

edges.

2.11 Summary

Concept	**Key Idea**
Graph	$G = (V, E)$
Directed graph	Edges have direction
Undirected graph	Edges have no direction
Self-loop	Edge from a vertex to itself
Multiple edges	Multiple edges between the same pair
Simple graph	No self-loops or multiple edges
Subgraph	$V' \subseteq V,\ E' \subseteq E$
Induced subgraph	Contains all edges among the selected vertices
Path	Sequence of connected vertices without repetition
Connected graph	A path exists between every pair of vertices
Cycle	Closed path with no other repeated vertex
Strongly connected	A directed path exists between every pair
Tree	Connected and acyclic
Tree edges	$|E| = |V| - 1$