CS202 – Data Structures

LECTURE-04

More on Asymptotic Analysis

Examples of Big-O, Big-Omega, Big-Theta, List ADT

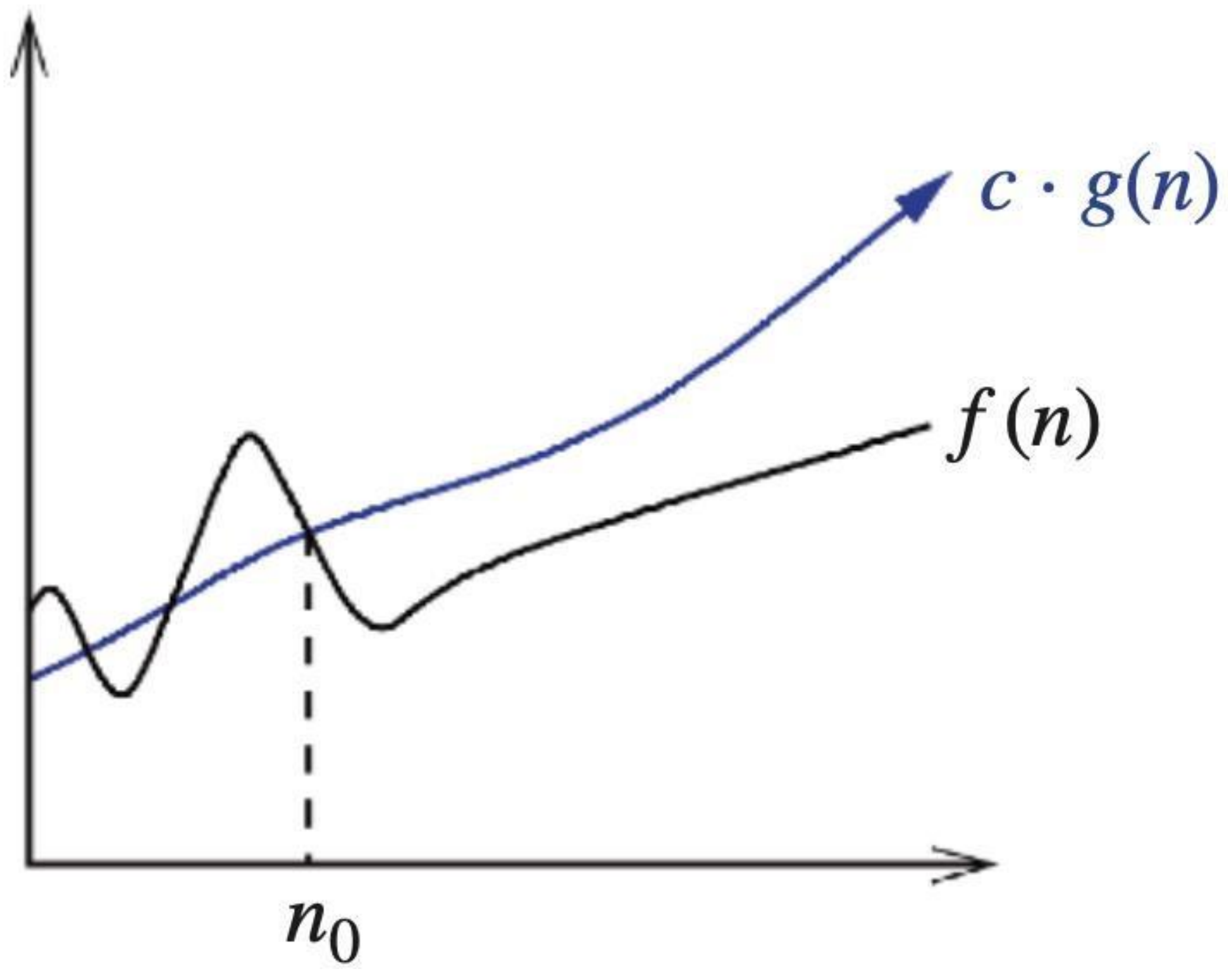
Dr. Maryam Abdul Ghafoor
Assistant Professor
Department of Computer Science, SBASSE

Agenda

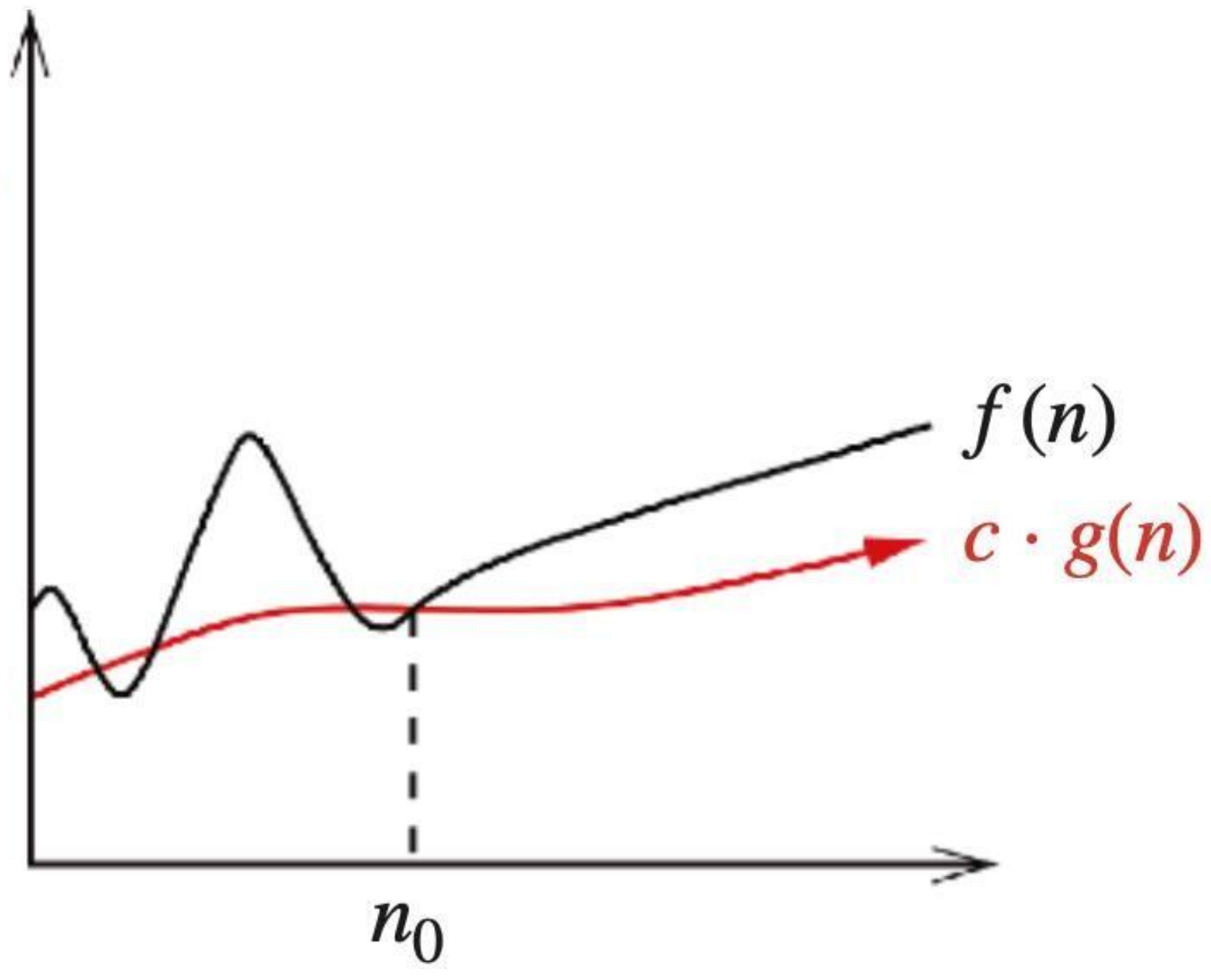
- Upper, Lower and Tight Bound
- Abstract Data Types versus Data Structures

Asymptotic Notations

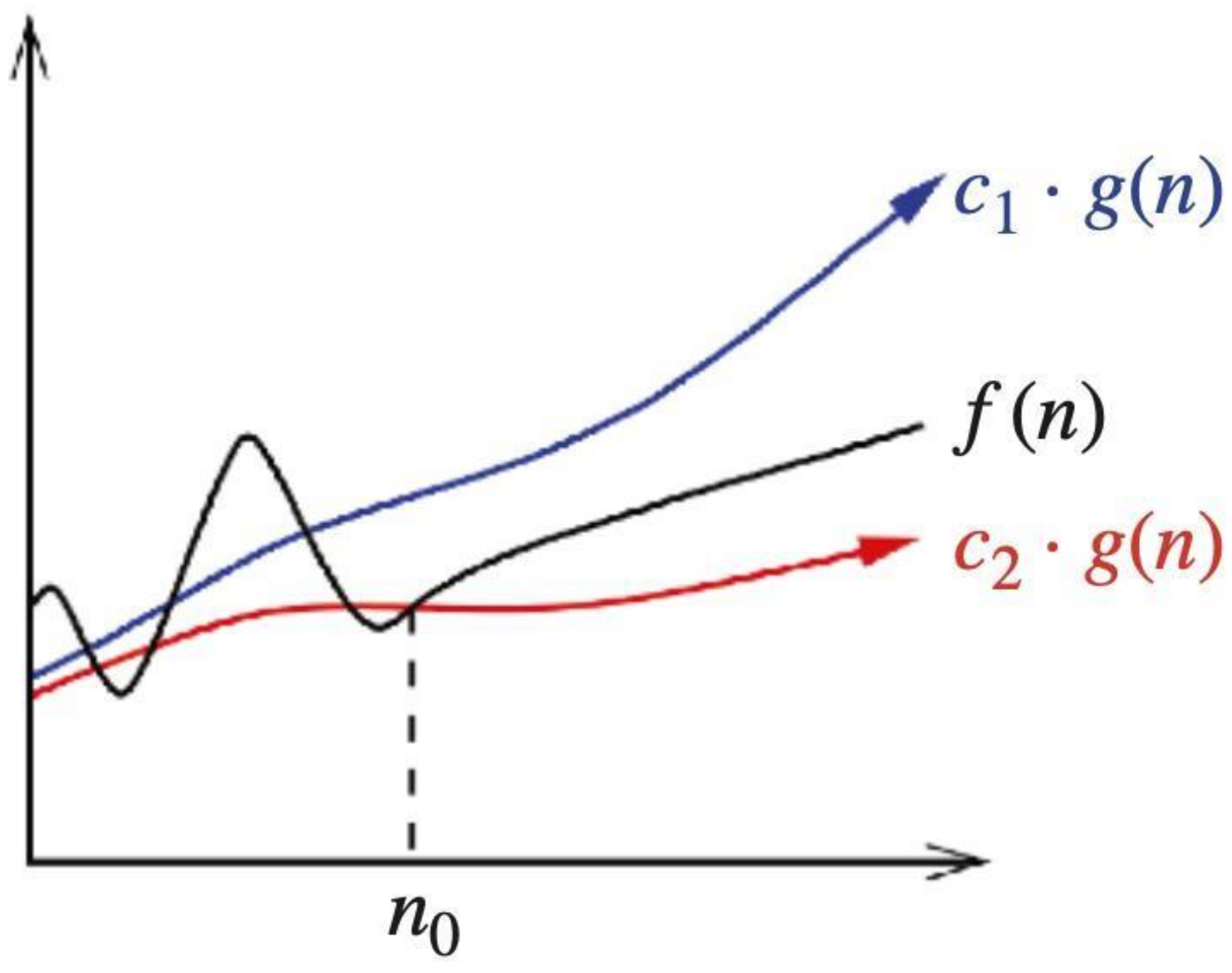
- Big-O: O(.)
- Big-Omega: Ω(.)
- Big-Theta: Θ(.)



$f(n) \in O(g(n))$

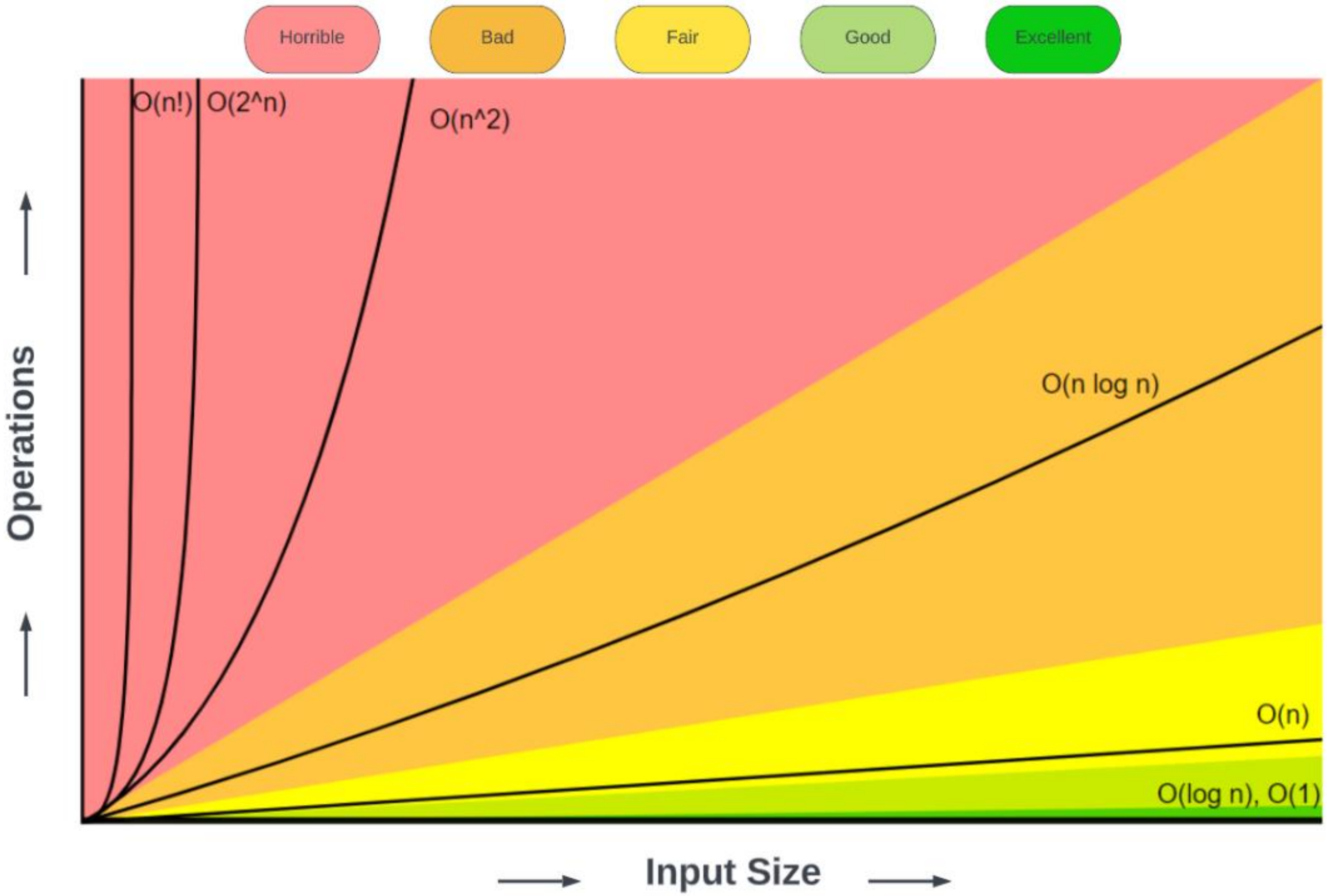


$f(n) \in \Omega(g(n))$



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

Big-O Complexity Chart for Common Functions



https://www.bigocheatsheet.com

Let's Practice

$8n^3 + 20n^2 + 50n + 100 \in \Omega(n)$

$f(n)$ is in $\Omega(g(n))$ if there exists positive constants c, n_o such that for all $n \geq n_o$

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

Let's Practice

Is $5n^2 \in \theta(n)$?

$f(n)$ is in $\Theta\left(g(n)\right)$, if there exists positive constants c, n_o such that for all $n \geq n_o$

$$f(n) \leq c.\mathrm{O}\left(g(n)\right)$$

and

$$f(n) \geq c.\Omega\left(g(n)\right)$$

Let's Practice

- Is $n^2 \in \Theta(n \log n)$?

$f(n)$ is in $\Theta(g(n))$, if there exists positive constants c, n_o such that for all $n \geq n_o$

$$f(n) \leq c.O(g(n))$$

and

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

Let's Practice

- Is $2^n \in \Theta(n^3)$?

$f(n)$ is in $\Theta\big(g(n)\big)$, if there exists positive constants c, n_o such that for all $n \geq n_o$

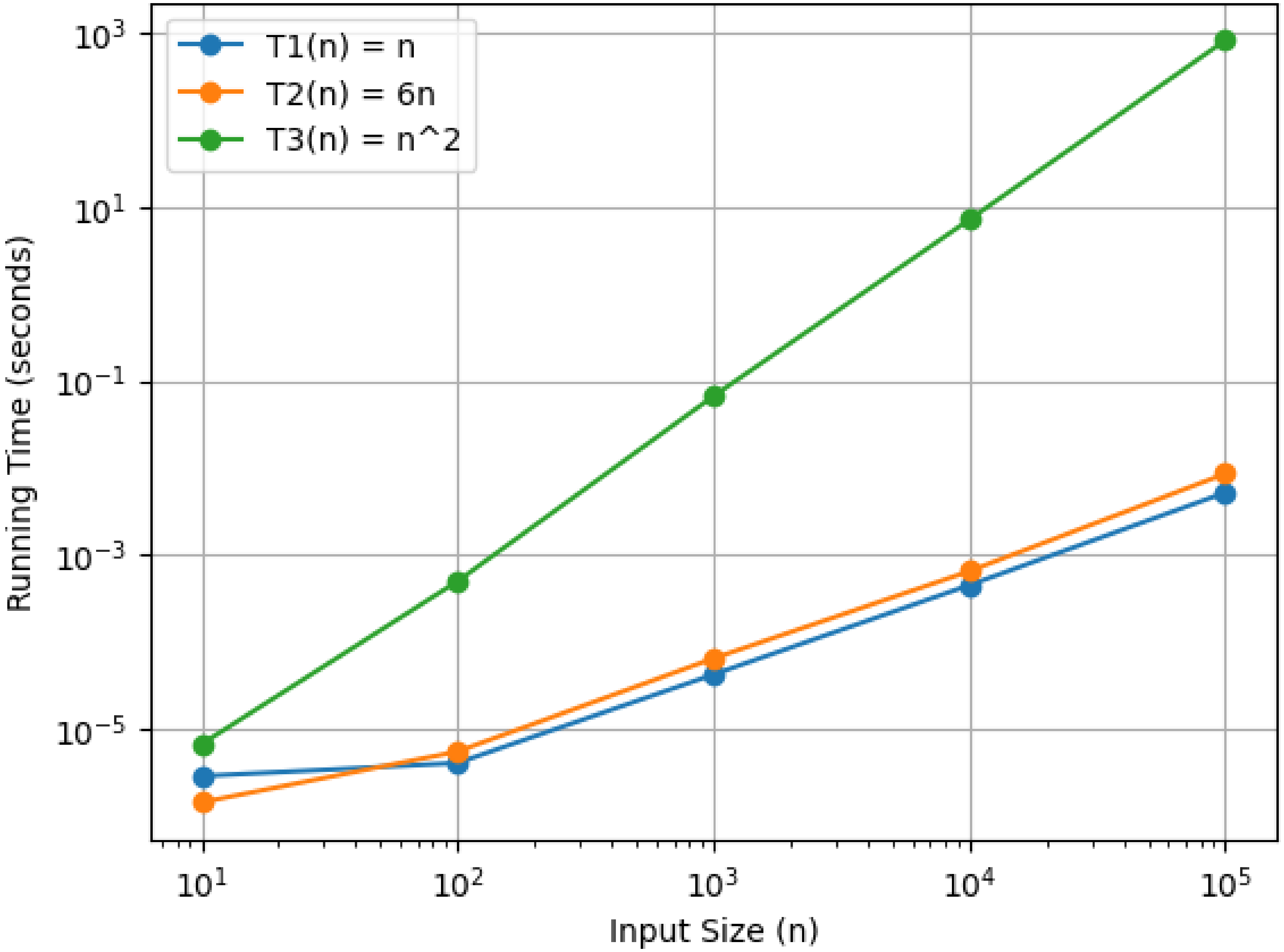
$$f(n) \leq c.\mathrm{O}\big(g(n)\big)$$

and

$$f(n) \geq c.\Omega\big(g(n)\big)$$

Limitation of Big-O Analysis

- Constants affect speed within the same complexity class
- Growth rate (Big-O) dominates for different complexity classes



Complexity Analysis

$\log_2 n$	n	$n \log_2 n$	n^2	2^n
2	**4**	8	16	16
3	**8**	24	64	256
4	**16**	64	256	65,536
5	**32**	160	1,024	4,294,967,296
6	**64**	384	4,096	1.84×10^{19}
7	**128**	896	16,384	3.40×10^{38}
8	**256**	2,048	65,536	1.16×10^{77}
9	**512**	4,608	262,144	1.34×10^{154}
10	**1,024**	10,240	1,048,576	1.80×10^{308}
30	**~1.07×10^9**	**32,212,254,720**	**1.15×10^{18}**	**21,073,741,824**

If n = 1GB, and each operation takes 1 μs, then

30sec	**~17.9 Min**	**~8.95 hours**	**~36,500 years**	**> 36,500 years**

Questions