BIG – O

zyBooks Chapter: 9.2

FUNCTIONS/ALGORITHMS ANALYSIS

Space Complexity

How much storage/memory they need to run

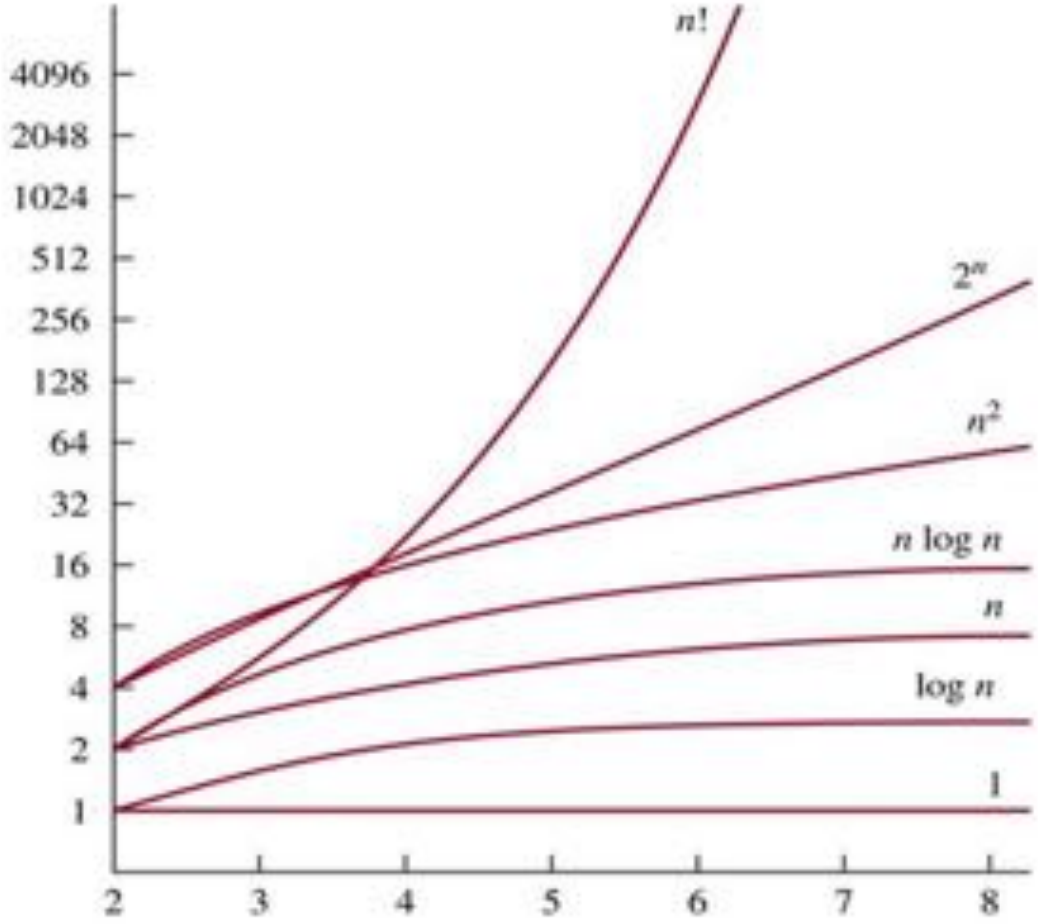
Time Complexity

How much time they take to run

===== SIZE OF INPUT =====

The Growth of Combinations of Functions

- 1
- $\log n$
- n
- $n \log n$
- n^2
- 2^n
- $n!$

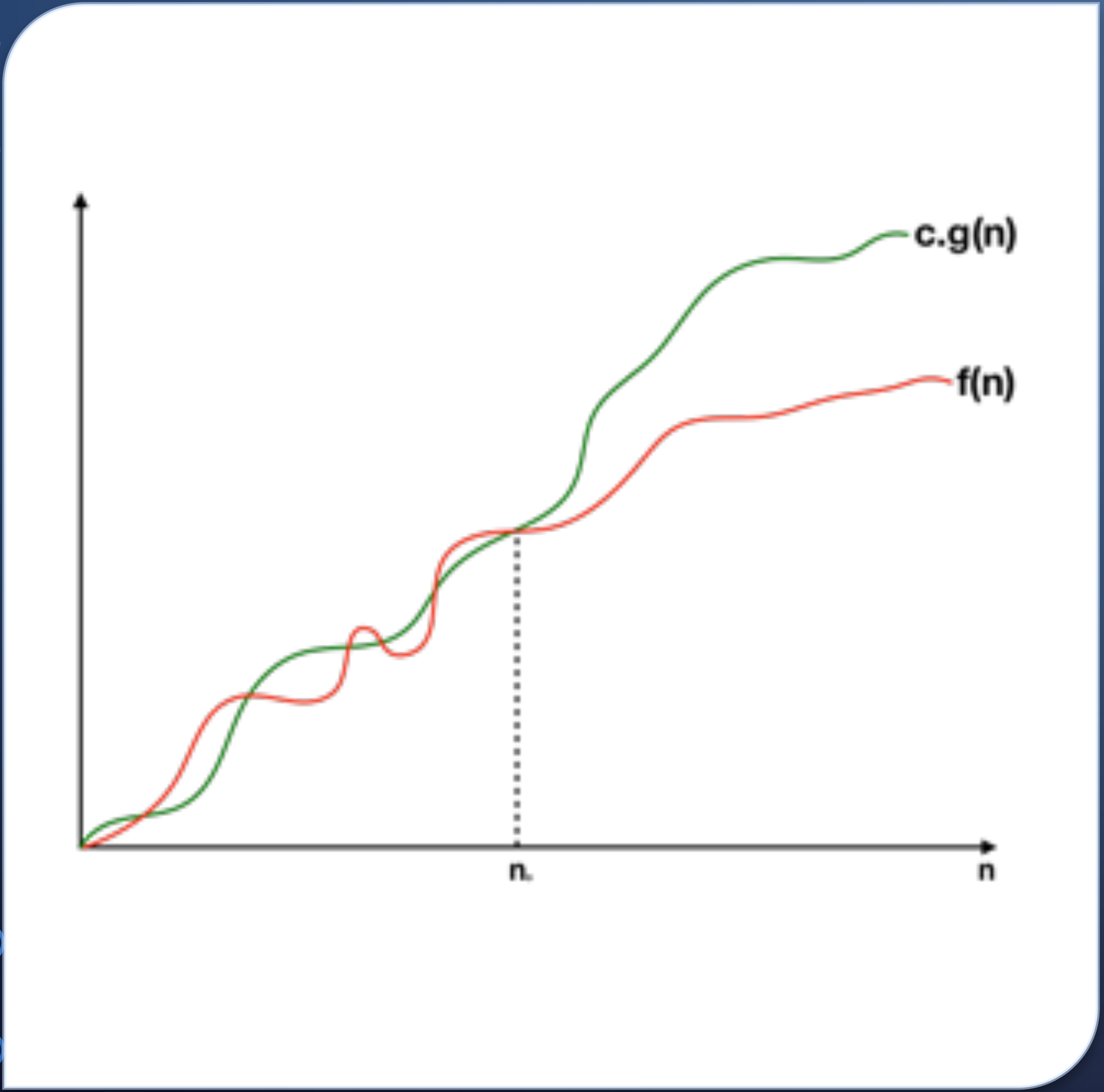


GROWTH OF FUNCTIONS

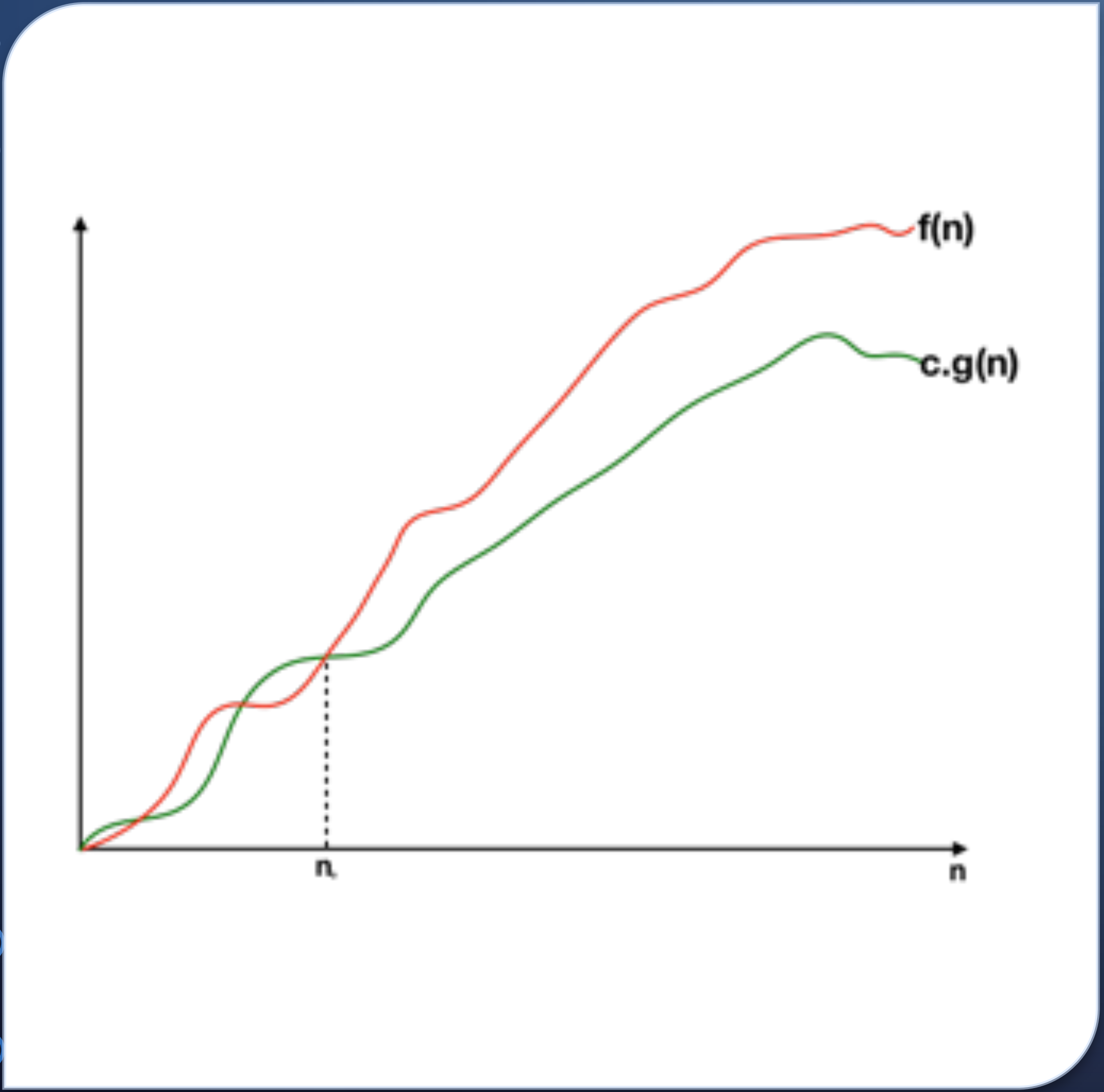
- Big – O
- Big – Ω
- Big – Θ

BIG – O

Upper Bound



- $O(\ g(n)\ )$: set of functions which grow no faster than g(n).
- $f(n) \in O(\ g(n)\ )\ \ iff$
  $\exists c \exists n_0\ \forall n \geq n_0\ \ f(n) \leq c \cdot g(n)$



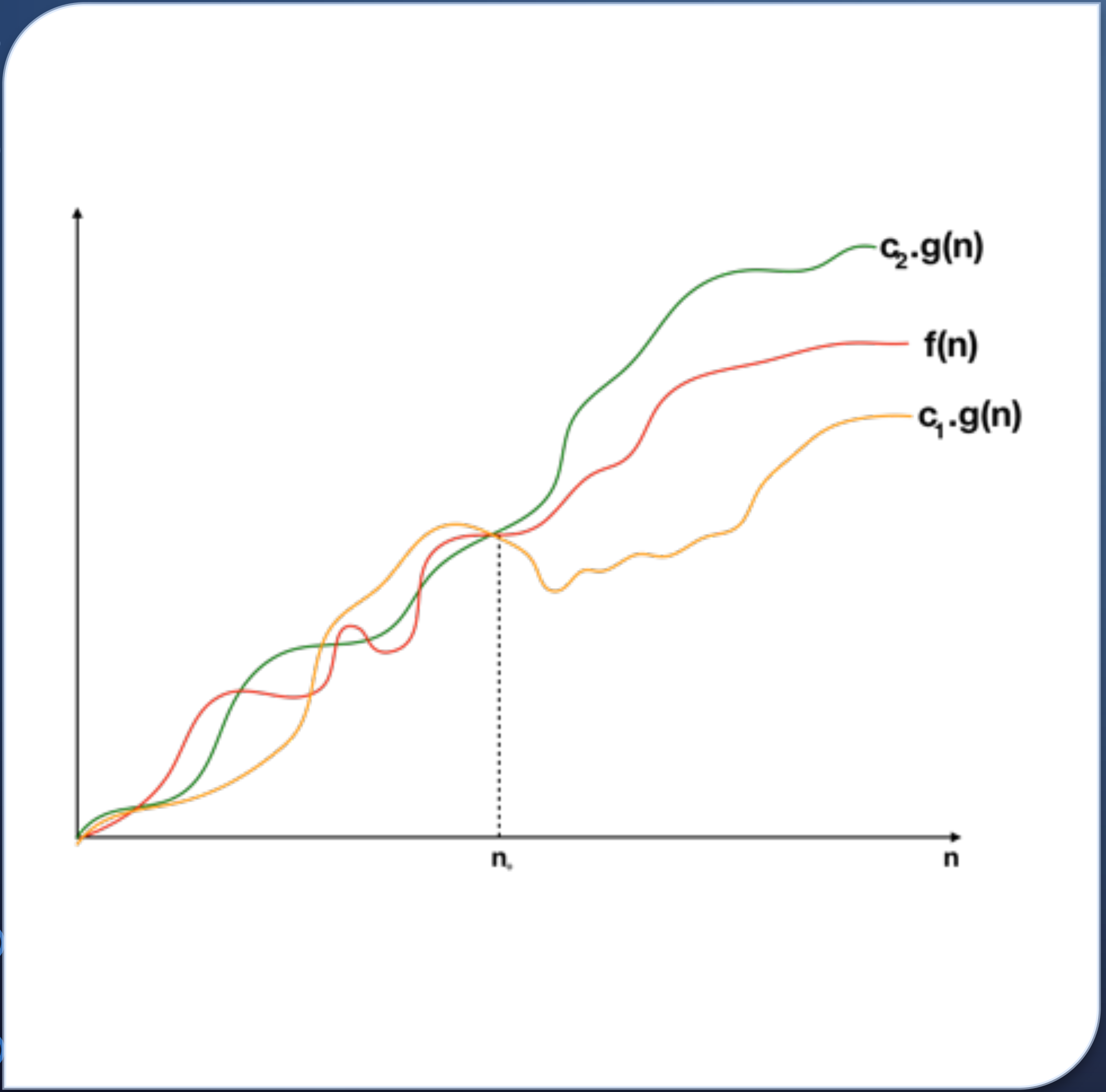
BIG – Ω

Lower Bound

- $\Omega(\ g(n)\ )$: set of functions which grow no slower than g(n).
- $f(n) \in \Omega(\ g(n)\ )\ \ iff$
  $\exists c \exists n_0\ \forall n \geq n_0\ \ f(n) \geq c \cdot g(n)$

Tight Bound

BIG – Θ



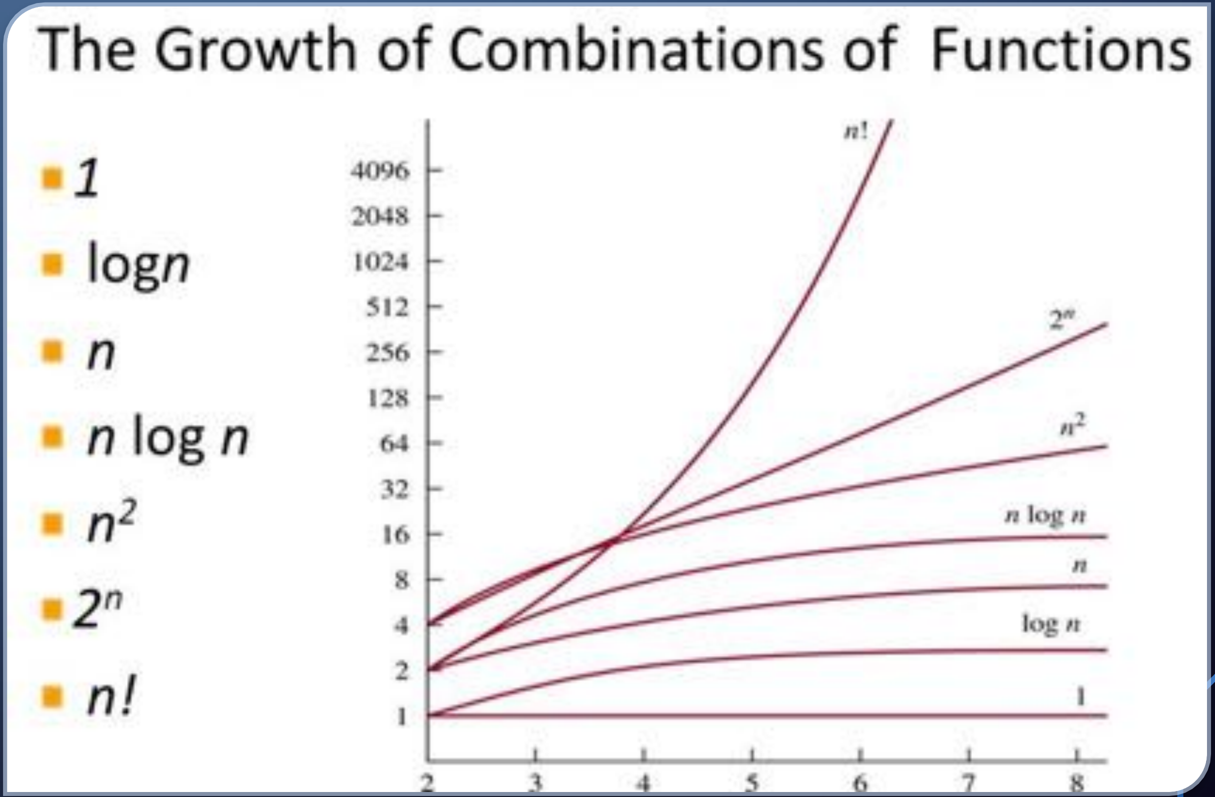
- $\Theta(\ g(n)\ )$: set of functions which grow at the same rate as g(n).
- $f(n) \in \Theta(\ g(n)\ )\ \ iff$
  $\exists c \exists n_0\ \forall n \geq n_0$
  $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$

COMPLEXITY PROBLEMS

- Given a function/algorithm, find its Big – O or Big – Ω
- Given two functions/algorithms, prove whether one is Big – O / Big – Ω / Big – Θ of the other

STEPS TO FIND BIG – O OF A FUNCTION

1) Find the dominant term (as a function of the variable)
2) Replace all other functions with the variable with 1)
3) Simplify
4) Remove constants



$$n! > 2^n > n^2 > n\log n > n > \log n > 1$$

EXAMPLE

1) Find the dominant term (as a function of the variable)
2) Replace all other functions with the variable with 1)
3) Simplify
4) Remove constants

$n! > 2^n > n^2 > n \log n > n > \log n > 1$

Q: Find the Big – O of $f(n) = 2n^2 + n + 2$

1) n^2 is the dominant term
2) $2n^2 + n + 2 \leq 2n^2 + n^2 + 2n^2$
3) $2n^2 + n + 2 \leq 5n^2 \rightarrow f(n) \leq 5n^2$

$\exists c \exists n_0 \forall n \geq n_0 f(n) \leq c \cdot g(n)$

4) So we have $c = 5$, and $g(n) = n^2$

When would this inequality be true?

$n \geq 1$

That is, $n_0 = 1$

Therefore, $f(n) \in O(n^2)$

SOME BIG – O PROPERTIES

$n! > 2^n > n^2 > n\log n > n > \log n > 1$

- When adding functions, the Big – O of the fastest growing function dominates
  - E.g., $3n\log n + 2000\log n + 20n + 10000 \in O(n\log n)$
- When multiplying functions, the Big – O of the functions are multiplied
  - E.g., $n^3(n^2\log n + n)$
  - $n^3 \in O(n^3)$
  - $n^2\log n + n \in O(n^2\log n)$
  - Therefore, $n^3(n^2\log n + n) \in O(n^3 \cdot n^2\log n) = O(n^5\log n)$

Q: Find the Big – O of $f(n) = 3n + 1$

1) n is the dominant term

2) 3n + 1 $\leq$ 3n+ n

3) 3n + 1 $\leq$ 4n → $f(n) \leq 4n$

$\exists c \exists n_0 \forall n \geq n_0 f(n) \leq c \cdot g(n)$

So we have c = 4, and g(n) = n

When would this inequality be true?

$n \geq 1$

That is, n_0 = 1

Therefore, $f(n) \in O(n)$

$n! > 2^n > n^2 > n \log n > n > \log n > 1$

1) Find the dominant term (as a function of the variable)

2) Replace all other functions with the variable with 1)

3) Simplify

4) Remove constants

Q: Find the Big – O of

$$f(n) = n\left(n^3(\log n + 2) + \log(5n^{12} + 24n) + n(5n^2 + 1) \right)$$

Hint: find the Big – O of the addition part first.

$n! > 2^n > n^2 > n \log n > n > \log n > 1$

$$n^3(\log n + 2) = n^3 \log n + 2n^3 \in O(n^3 \log n)$$

$$\log(5n^{12} + 24n) \in O(\log(n^{12}))$$

$$n(5n^2 + 1) = 5n^3 + n \in O(n^3)$$

The dominant one is $O(n^3 \log n)$

We know n $\in O(n)$. Then we multiply the Big – O's.

Therefore, $f(n) \in O(n^4 \log n)$

Q: Prove that $x^3 + 5x + 10 \in \Omega(x^2)$

$\Omega(\ g(n)\ )$: set of functions which grow no slower than g(n), and

$$\exists c \exists n_0\ \forall n \geq n_0\ \ f(n) \geq c \cdot g(n)$$

So we have the inequality

$$x^3 + 5x + 10 \geq x^2$$

$$x^3 + 5x + 10 \geq x^3 \geq x^2\ when\ x \geq 1$$

Therefore, $c = 1, x_0 = 1$.

Q: Prove that $3x^3 + 3x + 3 \in \Theta(x^3)$

Case 1: $O(x^3)$

We have the inequality

$$3x^3 + 3x + 3 \le x^3$$

$$3x^3 + 3x + 3 \le 3x^3 + 3x^3 + 3x^3 \quad when\ x \ge 1$$

$$3x^3 + 3x + 3 \le 9x^3\ when\ x \ge 1$$

Therefore, $c = 9, x_0 = 1$

Case 2: $\Omega(x^3)$

We have the inequality

$3x^3 + 3x + 3 \ge x^3\ \ when\ x \ge 1$ (to make the Big – O case true as well)

Therefore, $c = 1, x_0 = 1$.

After-Class Exercise: Select all that apply

$13 \log n$

$O(n)$	$O(n^2)$	$O(n \log n)$
$\Omega(n)$	$\Omega(n^2)$	$\Omega(n \log n)$
$\Theta(n)$	$\Theta(n^2)$	$\Theta(n \log n)$

$27n! + 2^n$

$O(n!)$	$O(2^n)$
$\Omega(n!)$	$\Omega(2^n)$
$\Theta(n!)$	$\Theta(2^n)$

$n(\log n + n^2) + n^2 - 1$

$O(n^3)$	$O(n)$	$O(n \log n)$
$\Omega(n^3)$	$\Omega(n)$	$\Omega(n \log n)$

ADDITIONAL READING MATERIAL

Growth of a Function

https://www.codesdope.com/course/algorithms-growth-of-a-function/

Solution to the After-Class Exercise

$13 \log n$

$O(n)$ ✓	$O(n^2)$ ✓	$O(n \log n)$ ✓
$\Omega(n)$	$\Omega(n^2)$	$\Omega(n \log n)$
$\Theta(n)$	$\Theta(n^2)$	$\Theta(n \log n)$

$27n! + 2^n$

$O(n!)$ ✓	$O(2^n)$
$\Omega(n!)$ ✓	$\Omega(2^n)$ ✓
$\Theta(n!)$ ✓	$\Theta(2^n)$

$n(\log n + n^2) + n^2 - 1$

$O(n^3)$ ✓	$O(n)$	$O(n \log n)$
$\Omega(n^3)$ ✓	$\Omega(n)$ ✓	$\Omega(n \log n)$ ✓