

# Big O Notation

# Runtime of an algorithm

Running time of an algorithm depends on

i) Input size

6	13	14	25	33
---	----	----	----	----

Array size = 5

6	13	14	25	33	43	51	53	64	72	84
---	----	----	----	----	----	----	----	----	----	----

Array size = 11

# Runtime of an algorithm

Running time of an algorithm depends on

i) Input size

a function of the size of its input

$f(n)$

$n$  = input size

6	13	14	25	33
---	----	----	----	----

Array size = 5

6	13	14	25	33	43	51	53	64	72	84
---	----	----	----	----	----	----	----	----	----	----

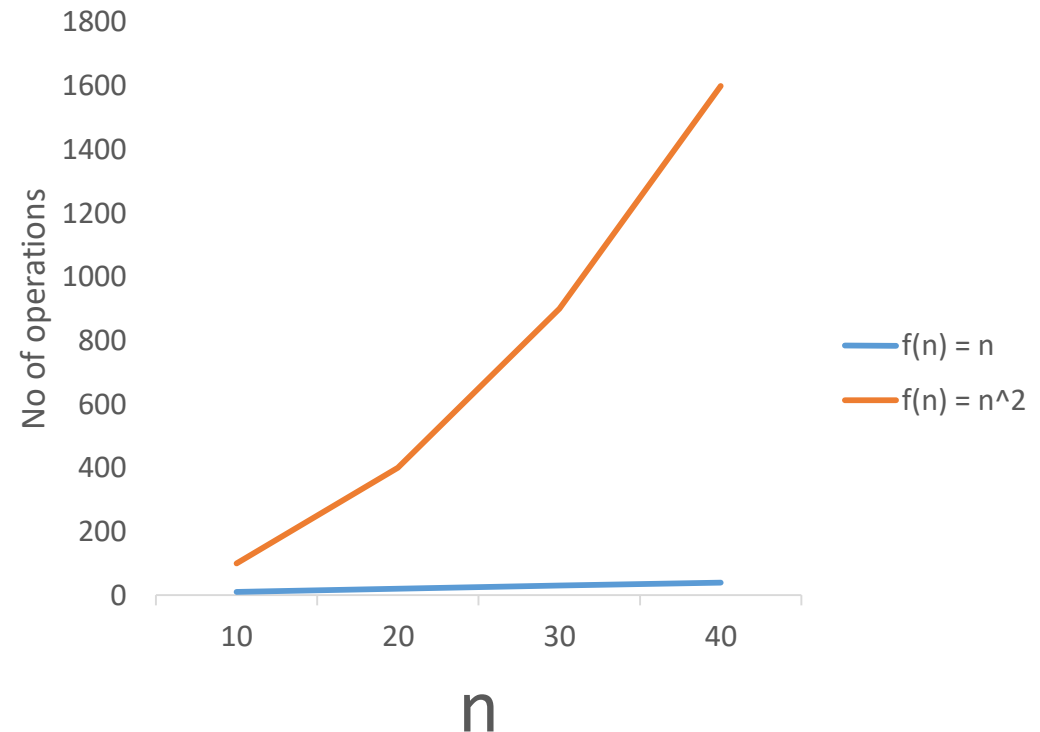
Array size = 11

# Growth rate of a function

**Rate of growth:** How fast a function grows with the input size

$$f(n) = n$$

$$f(n) = n^2$$

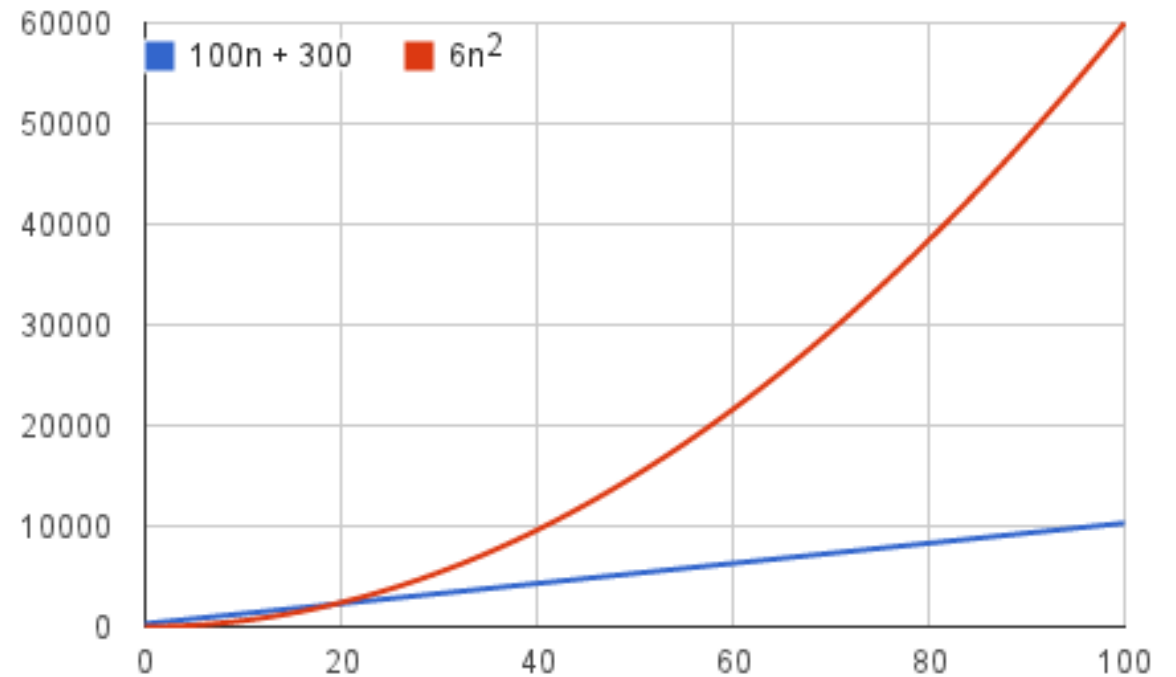


# Asymptotic notation of a function

$$f(n) = 6n^2 + 100n + 30$$

$$n \geq 20$$

$$6n^2 > 100n + 30$$

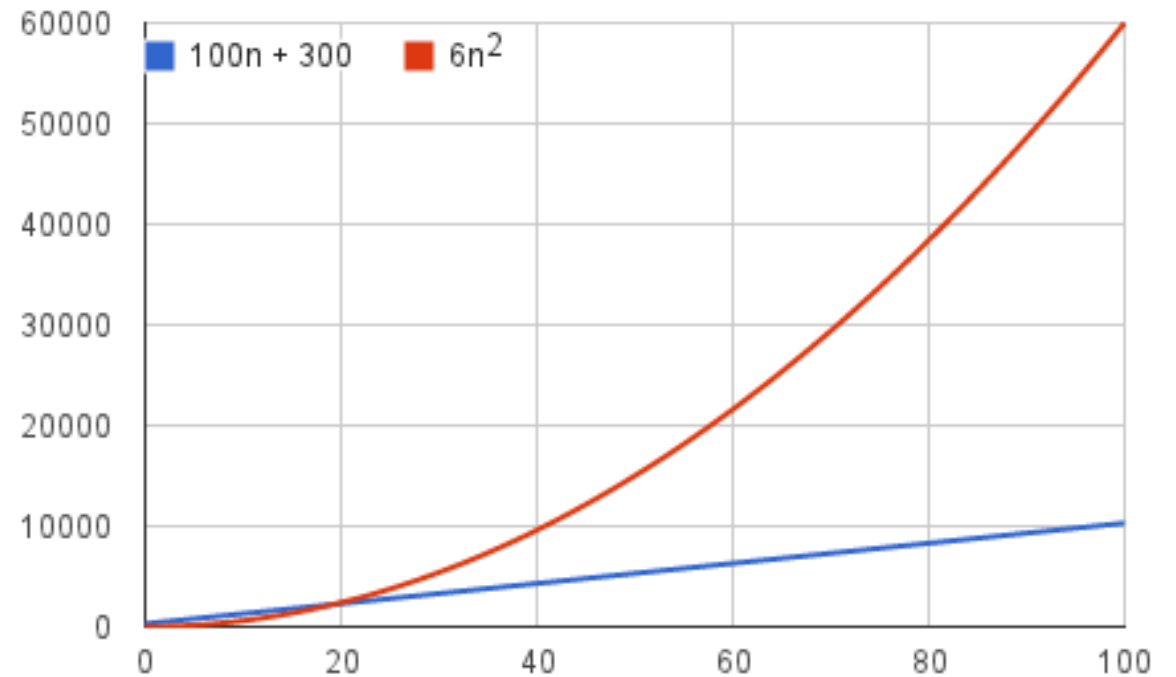


# Asymptotic notation of a function

$$f(n) = 6n^2 + 100n + 30$$

The algorithm grows as  $n^2$

Drop coefficient 6 and terms  $100n + 30$ . They are not significant enough.

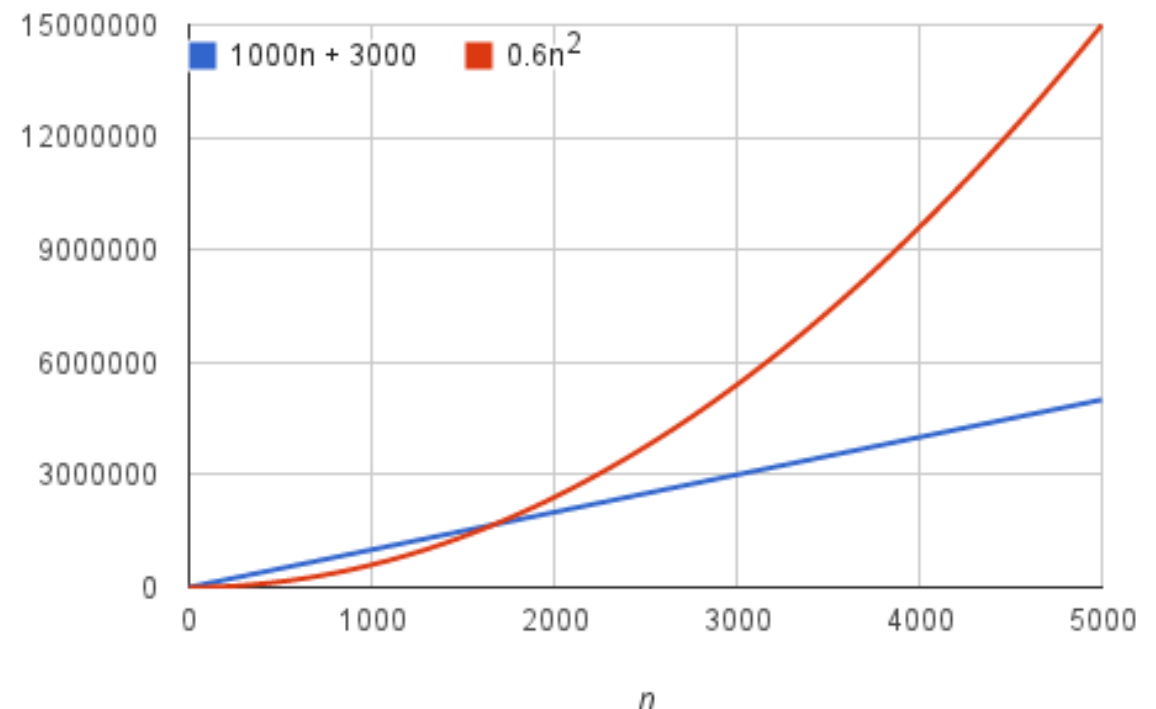


# Asymptotic notation of a function

$$f(n) = 0.6n^2 + 1000n + 3000$$

$$n \geq 1000$$

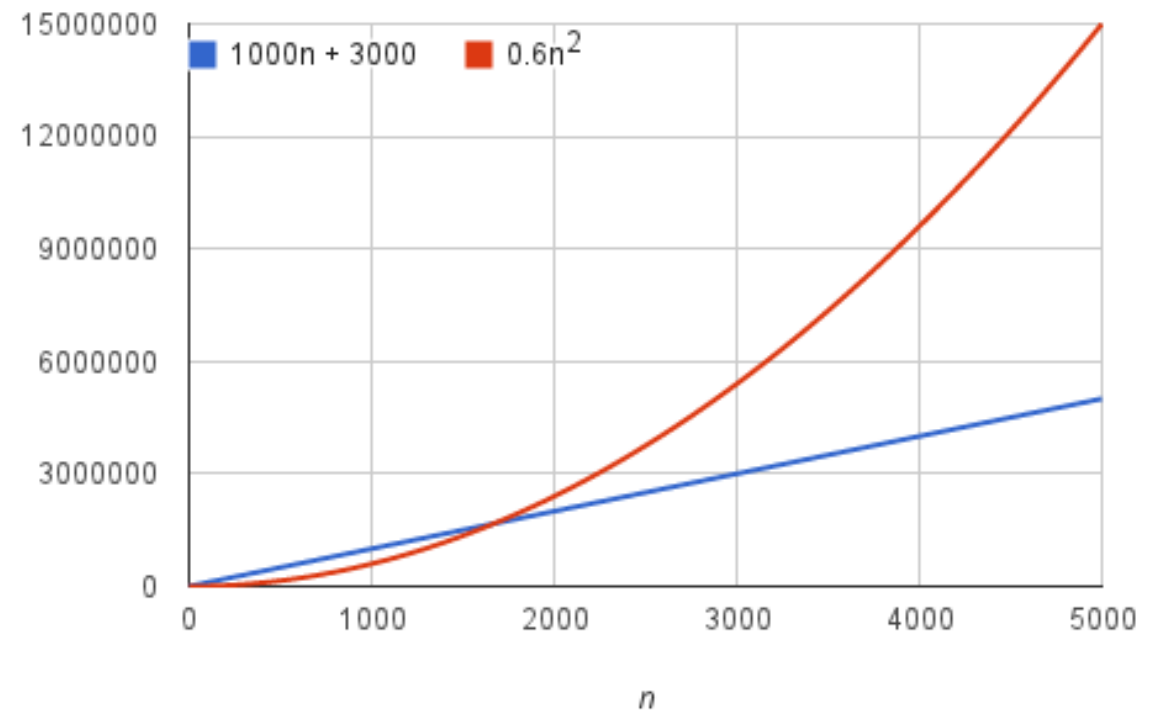
$$0.6n^2 > 1000n + 3000$$



# Asymptotic notation of a function

$$f(n) = 0.6n^2 + 1000n + 3000$$

The algorithm grows as  $n^2$





# Asymptotic notation of a function

$$f(n) = 0.6n^2 + 1000n + 3000$$

$$f(n) = 6n^2 + 100n + 30$$

These algorithm grows as  $n^2$

When we drop the constant coefficients and the less significant terms, we use **asymptotic notation**

# Big O Notation

Given  $f(n)$ : the actual growth rate of your algorithm as a function of input size find  $g(n)$  such that

$$C|g(n)| \geq |f(n)| \text{ for } n > k$$

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

# Big O Notation

Find the Big O Notation of  
 $n^2 + 2n + 1$

$$\begin{aligned} f(n) = n^2 + 2n + 1 &\leq n^2 + 2n^2 + n^2 \\ &= 4n^2 = Cg(n) \end{aligned}$$

$$f(n) = O(g(n)) = O(n^2),$$

Where  $k = 1$ ,  $C = 4$

Find k:

$$n = 1, f(n) = 4, Cg(n) = 4$$

$$n = 2, f(n) = 9, Cg(n) = 16$$

For  $n > 1$   $Cg(n) \geq f(n)$

Hence,  $k = 1$

# Big O Notation

Find the Big O Notation of  $n!$

$$\begin{aligned} n! &= 1.2.3 \dots n \leq n.n \dots n \\ &= n^n = C g(n) \end{aligned}$$

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

Where  $k = 1$ ,  $C = 1$

Find  $k$ :

$$n = 1, f(n) = 1, Cg(n) = 1$$

$$n = 2, f(n) = 2, Cg(n) = 4$$

For  $n > 1$   $Cg(n) \geq f(n)$

Hence,  $k = 1$

# Big O Notation

Find the Big O Notation of  $\log(n!)$

# Big O Notation

If  $f_1(x) = O(g_1(x))$  and  $f_2(x) = O(g_2(x))$

$$(f_1 + f_2)(x) = O(\max(g_1(x), g_2(x)))$$

$$(f_1 f_2)(x) = O(g_1(x) g_2(x))$$

# Big O Notation

Find the Big O Notation of  $f(n) = 3n \log(n!) + (n^2 + 3) \log(n)$

# Big O Notation

Find the Big O Notation of  $f(n) = 3n \log(n!) + (n^2 + 3) \log(n)$



# Common Big O Notation

Notation	Name
$O(1)$	Constant
$O(\log n)$	Logarithmic
$O((\log n)^c)$	Poly-logarithmic
$O(n)$	Linear
$O(n^2)$	Quadratic
$O(n^c)$	Polynomial
$O(c^n)$	Exponential

