

## Big-O versus exact time complexity

- ▶ If the time complexity is  $5n^2 + 3n$  and we say it is  $O(n^2)$ , we:

# Big-O versus exact time complexity

- ▶ If the time complexity is  $5n^2 + 3n$  and we say it is  $O(n^2)$ , we:
  - ▶ Lose information about the exact number of basic operations for a given value of  $n$ .

## Big-O versus exact time complexity

- ▶ If the time complexity is  $5n^2 + 3n$  and we say it is  $O(n^2)$ , we:
  - ▶ Lose information about the exact number of basic operations for a given value of  $n$ .
  - ▶ Gain a simple way of expressing that the time complexity grows with the square of the input size.

## Big-O versus exact time complexity

- ▶ If the time complexity is  $5n^2 + 3n$  and we say it is  $O(n^2)$ , we:
  - ▶ Lose information about the exact number of basic operations for a given value of  $n$ .
  - ▶ Gain a simple way of expressing that the time complexity grows with the square of the input size.

$T(n) = 5n^2 + 3n$	$T(n) = O(n^2)$
What is the worst-case number of operations for $n = 1000$ ?	
If the algorithm takes at most 1 second for $n = 1000$ , how long will it take at most for $n = 2000$ ?	

## Big-O versus exact time complexity

- ▶ If the time complexity is  $5n^2 + 3n$  and we say it is  $O(n^2)$ , we:
  - ▶ Lose information about the exact number of basic operations for a given value of  $n$ .
  - ▶ Gain a simple way of expressing that the time complexity grows with the square of the input size.

$T(n) = 5n^2 + 3n$	$T(n) = O(n^2)$
What is the worst-case number of operations for $n = 1000$ ?	
5,003,000	
If the algorithm takes at most 1 second for $n = 1000$ , how long will it take at most for $n = 2000$ ?	

## Big-O versus exact time complexity

- ▶ If the time complexity is  $5n^2 + 3n$  and we say it is  $O(n^2)$ , we:
  - ▶ Lose information about the exact number of basic operations for a given value of  $n$ .
  - ▶ Gain a simple way of expressing that the time complexity grows with the square of the input size.

$T(n) = 5n^2 + 3n$	$T(n) = O(n^2)$
What is the worst-case number of operations for $n = 1000$ ?	
5,003,000	???
If the algorithm takes at most 1 second for $n = 1000$ , how long will it take at most for $n = 2000$ ?	

## Big-O versus exact time complexity

- ▶ If the time complexity is  $5n^2 + 3n$  and we say it is  $O(n^2)$ , we:
  - ▶ Lose information about the exact number of basic operations for a given value of  $n$ .
  - ▶ Gain a simple way of expressing that the time complexity grows with the square of the input size.

$T(n) = 5n^2 + 3n$	$T(n) = O(n^2)$
What is the worst-case number of operations for $n = 1000$ ?	
5,003,000	???
If the algorithm takes at most 1 second for $n = 1000$ , how long will it take at most for $n = 2000$ ?	
$\frac{20006000}{5003000} \times 1 \text{ sec} = 3.9988 \text{ secs}$	

## Big-O versus exact time complexity

- ▶ If the time complexity is  $5n^2 + 3n$  and we say it is  $O(n^2)$ , we:
  - ▶ Lose information about the exact number of basic operations for a given value of  $n$ .
  - ▶ Gain a simple way of expressing that the time complexity grows with the square of the input size.

$T(n) = 5n^2 + 3n$	$T(n) = O(n^2)$
What is the worst-case number of operations for $n = 1000$ ?	
5,003,000	???
If the algorithm takes at most 1 second for $n = 1000$ , how long will it take at most for $n = 2000$ ?	
$\frac{20006000}{5003000} \times 1 \text{ sec} = 3.9988 \text{ secs}$	$4 \times 1 \text{ sec} = 4 \text{ secs}$