Time Complexity

- Describe an algorithm's running time in terms of its asymptotic running time (how fast the time increases as the input gets large)

- Time complexity classes

$O(1)$	$O(n^2)$
$O(\lg n)$	$O(n^3)$
$O(n)$	$\vdots$
$O(n \lg n)$	$O(2^n)$

O - asymptotic upper bound

Ω - asymptotic lower bound

Θ - both upper and lower bound

Insertion sort vs quicksort

$[1, 2, 3, 4]$

Insertion sort	quicksort
good for small arrays	good for large arrays
Worst case $\Theta(n^2)$	worst case $\Theta(n^2)$
Best case $\Theta(n)$	expected worst case with random pivot $O(n \lg n)$
Good for almost sorted arrays	best case $\Theta(n \lg n)$

both in-place

both comparison sorts

Counting sort

Requires extra space

Requires integers with a known range

If k possible distinct values and $k = O(n)$, time is $\Theta(n)$