

# Algorithms Summary

# Algorithms Summary

- There are many different types of searching and sorting algorithms, and we'll only skim the surface in CS50.
- Let's see if we can't distill each algorithm down to its core elements, so you can remember the most important information about each.

# Algorithms Summary

Algorithm Name	Basic Concept	$O$	$\Omega$
Selection Sort	Find the <b>smallest</b> unsorted element in an array and swap it with the <b>first</b> unsorted element of that array.	$n^2$	$n^2$

# Algorithms Summary

Algorithm Name	Basic Concept	$O$	$\Omega$
Selection Sort	Find the <b>smallest</b> unsorted element in an array and swap it with the <b>first</b> unsorted element of that array.	$n^2$	$n^2$
Bubble Sort	Swap <b>adjacent pairs</b> of elements if they are out of order, effectively “bubbling” larger elements to the right and smaller ones to the left.	$n^2$	$n$

# Algorithms Summary

Algorithm Name	Basic Concept	$O$	$\Omega$
Selection Sort	Find the <b>smallest</b> unsorted element in an array and swap it with the <b>first</b> unsorted element of that array.	$n^2$	$n^2$
Bubble Sort	Swap <b>adjacent pairs</b> of elements if they are out of order, effectively “bubbling” larger elements to the right and smaller ones to the left.	$n^2$	$n$
Insertion Sort	Proceed once through the array from left-to-right, <b>shifting</b> elements as necessary to insert each element into its correct place.	$n^2$	$n$

# Algorithms Summary

Algorithm Name	Basic Concept	$O$	$\Omega$
Selection Sort	Find the <b>smallest</b> unsorted element in an array and swap it with the <b>first</b> unsorted element of that array.	$n^2$	$n^2$
Bubble Sort	Swap <b>adjacent pairs</b> of elements if they are out of order, effectively “bubbling” larger elements to the right and smaller ones to the left.	$n^2$	$n$
Insertion Sort	Proceed once through the array from left-to-right, <b>shifting</b> elements as necessary to insert each element into its correct place.	$n^2$	$n$
Merge Sort	<b>Split</b> the full array into subarrays, then <b>merge</b> those subarrays back together in the correct order.	$n \log n$	$n \log n$

# Algorithms Summary

Algorithm Name	Basic Concept	$O$	$\Omega$
Selection Sort	Find the <b>smallest</b> unsorted element in an array and swap it with the <b>first</b> unsorted element of that array.	$n^2$	$n^2$
Bubble Sort	Swap <b>adjacent pairs</b> of elements if they are out of order, effectively “bubbling” larger elements to the right and smaller ones to the left.	$n^2$	$n$
Insertion Sort	Proceed once through the array from left-to-right, <b>shifting</b> elements as necessary to insert each element into its correct place.	$n^2$	$n$
Merge Sort	<b>Split</b> the full array into subarrays, then <b>merge</b> those subarrays back together in the correct order.	$n \log n$	$n \log n$
Linear Search	<b>Iterate</b> across the array from left-to-right, trying to find the target element.	$n$	$1$

# Algorithms Summary

Algorithm Name	Basic Concept	$O$	$\Omega$
Selection Sort	Find the <b>smallest</b> unsorted element in an array and swap it with the <b>first</b> unsorted element of that array.	$n^2$	$n^2$
Bubble Sort	Swap <b>adjacent pairs</b> of elements if they are out of order, effectively “bubbling” larger elements to the right and smaller ones to the left.	$n^2$	$n$
Insertion Sort	Proceed once through the array from left-to-right, <b>shifting</b> elements as necessary to insert each element into its correct place.	$n^2$	$n$
Merge Sort	<b>Split</b> the full array into subarrays, then <b>merge</b> those subarrays back together in the correct order.	$n \log n$	$n \log n$
Linear Search	<b>Iterate</b> across the array from left-to-right, trying to find the target element.	$n$	$1$
Binary Search	Given a <u>sorted</u> array, <b>divide and conquer</b> by systematically eliminating half of the remaining elements in the search for the target element.	$\log n$	$1$