Binary Search—Complexity Class: O(log N)

* Only works if the list is sorted

1. Compare the element at the middle position in the list to the target value.
2. If the target value is equal to the element at the middle position, then you are done.
3. If the target value is less than the element at the middle position, then repeat the procedure starting from step 1 for the left half of the list.
4. If the target value is greater than the element at the middle position, then repeat the procedure starting from step 1 for the right half of the list.

Note: If either the left or right sides of the list are empty for steps 3 or 4, then the target value is not contained in the list.

Target Value: 9

Index	0	1	2	3	4	5	6	7	8
	-3	6	9	12	15	18	21	24	27
	↑ LOW				↑ MID				↑ HIGH

Since 9 is less than 15 and the list is sorted, we know 9 can't possibly be in the second half of the list.
So we only continue searching in the first half.

Index	0	1	2	3	4	5	6	7	8
	-3	6	9	12	15	18	21	24	27
	↑ LOW	↑ MID		↑ HIGH					

9 is greater than 6, so we continue searching in the second half of the list.

Index	0	1	2	3	4	5	6	7	8
	-3	6	9	12	15	18	21	24	27
			↑ LOW ↑ MID	↑ HIGH					

We found 9!

Sorting Algorithms:

	Complexity	The Steps	Visual Representation
Selection	$O(N^2)$	1. Look through the entire list for the smallest value. 2. Swap the smallest value with the value at the current index (Unless current index contains the smallest value). 3. Increase current index. 4. Look through the rest of the list for the smallest value. 5. Swap this value with the value at current index. 6. Repeat for the rest of the list.	(Shaded boxes indicate swapped values)

Index	0	1	2	3	4	5	6	7	8
	7	4	2	16	22	13	15	31	0
	↑ Current Index								↑ Smallest Value

Index	0	1	2	3	4	5	6	7	8
	0	4	2	16	22	13	15	31	7
		↑ Current Index	↖ Smallest Value						

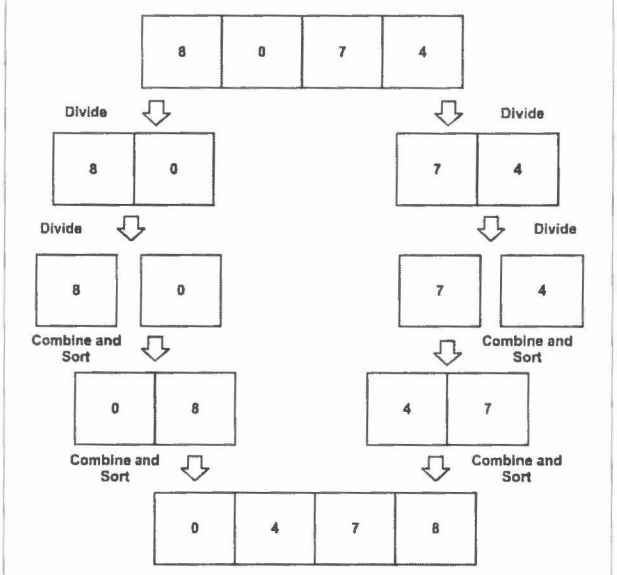
Index	0	1	2	3	4	5	6	7	8
	0	2	4	16	22	13	15	31	7
			↑ Current Index						

In this case, 4 is the smallest value so we don't need to swap anything.

Index	0	1	2	3	4	5	6	7	8
	0	2	4	16	22	13	15	31	7
				↑ Current Index					↑ Smallest Value

Index	0	1	2	3	4	5	6	7	8
	0	2	4	7	22	13	15	31	16

This process continues until you've reached the end of the list.

Mergesort	O(N log N)	1. Repeatedly divide the list into two equal parts until each part is a single element of the list 2. Combine the parts in sorted order, until the list is completely reconstructed.	

AP Computer Science A Searching and Sorting Algorithms Cheat Sheet

Insertion	O(N^2)	1. Divide list into two imaginary lists: sorted (initially empty) and unsorted (the rest of the elements). 2. Take the first element from the unsorted list and place into sorted. 3. Take the next element from the unsorted list and **insert** the value into the correct location. 4. Repeat until the unsorted part is empty.