|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm Name | Basic Concept | Worst Case O | Best Case Ω |
| Selection Sort | Find the **smallest** unsorted element in an array and swap it with the **first** unsorted element of that array. | **n** 2 | **n** 2 |
| Bubble Sort | Swap **adjacent pairs** 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, **shifting** elements as necessary to insert each element into its correct place. | **n** 2 | n |
| Merge Sort | **Split** the full array into subarrays, then **merge** those subarrays back together into the correct order. | n log n | n log n |
| Linear Search | **Iterate** across the array from left-to-right, trying to find the target element. | n | 1 |
| Binary Search | Given a sorted array, **divide and conquer** by systematically eliminating half of the remaining elements in the search for the target elements | log n | 1 |