Quick Sort is a divide-and-conquer algorithm that works by selecting a pivot element from the list and partitioning the other elements into two sublists, according to whether they are less than or greater than the pivot.

The sublists are then recursively sorted, and the results are combined to produce the final sorted list.

Here’s how the Quick Sort algorithm works:

- Choose a pivot element from the list. This can be done randomly, or by selecting the first, last, or middle element of the list.
- Reorder the list so that all elements less than the pivot come before it, and all elements greater than the pivot come after it. This is called partitioning.
- Recursively apply steps 1-2 to the sublists of elements less than and greater than the pivot.
- Combine the sorted sublists, along with the pivot, to obtain the final sorted list.

Example:

Consider the following unsorted list of integers: [5, 3, 8, 2, 1].

Step 1:

- Choose the pivot element. For this example, let’s choose the first element as the pivot: 5.

Step 2:

- Reorder the list so that all elements less than 5 come before it, and all elements greater than 5 come after it. The list becomes: [3, 2, 1, 5, 8].

Step 3:

- Recursively apply Quick Sort to the sublists [3, 2, 1] and [8].
    - For the sublist [3, 2, 1]:
        - Choose the pivot. Let's choose the first element, which is 3.
        - Reorder the sublist: [1, 2, 3].
    - For the sublist [8]:
        - Since it contains only one element, no further sorting is needed.

Step 4:

- Combine the sorted sublists [1, 2, 3] and [8], along with the pivot 5, to obtain the final sorted list: [1, 2, 3, 5, 8].

The list is now sorted in ascending order: [1, 2, 3, 5, 8].

Quick Sort has a time complexity of O(n log n) in the average and best cases, where 'n' is the number of elements in the list.

However, in the worst case (when the pivot selection is unbalanced), it can have a time complexity of O(n^2).

Nevertheless, Quick Sort is efficient for large lists and is often used in practice due to its good average-case performance.