

Pre Requisites:

- Basic JAVA syntax
- Recursion concepts

List of concepts involved :

- Introduction to divide and conquer strategy
- Merge sort Algorithm
- Working of merge sort algorithm
- time and space complexity analysis of merge sort algorithm
- Quick sort Algorithm
- working of Quick sort algorithm
- Time and space complexity analysis of quick sort algorithm
- Randomized quick sort algorithm

A divide and conquer algorithm is a strategy of solving a large problem by

- Breaking the problem into smaller sub-problems
- Solving the sub-problems, and
- Combining them to get the desired output.
- To use the divide and conquer algorithm, recursion is used.

Here are the steps involved:

- Divide: Divide the given problem into subproblems using recursion.
- Conquer: Solve the smaller subproblems recursively. If the subproblem is small enough, then solve it directly.
- Combine: Combine the solutions of the sub-problems that are part of the recursive process to solve the actual problem.

Merge sort :

Merge sort uses the divide and conquer approach to sort the elements. It is one of the most popular and efficient sorting algorithms. It divides the given list into two equal halves, calls itself for the two halves and then merges the two sorted halves. We have to define the merge() function to perform the merging.

The sub-lists are divided again and again into halves until the list cannot be divided further. Then we combine the pair of one element lists into two-element lists, sorting them in the process. The sorted two-element pairs are merged into the four-element lists, and so on until we get the sorted list.

Algorithm of merge sort :

In the following algorithm, **array** is the given array, **begin** is the starting element, and **end** is the last element of the array.

```
MERGE_SORT(array, begin, end)
if begin < end
set mid = begin + (end-begin)/2
MERGE_SORT(array, begin, mid)
MERGE_SORT(array, mid + 1, end)
MERGE (array, begin, mid, end)
end of if
END MERGE_SORT
```