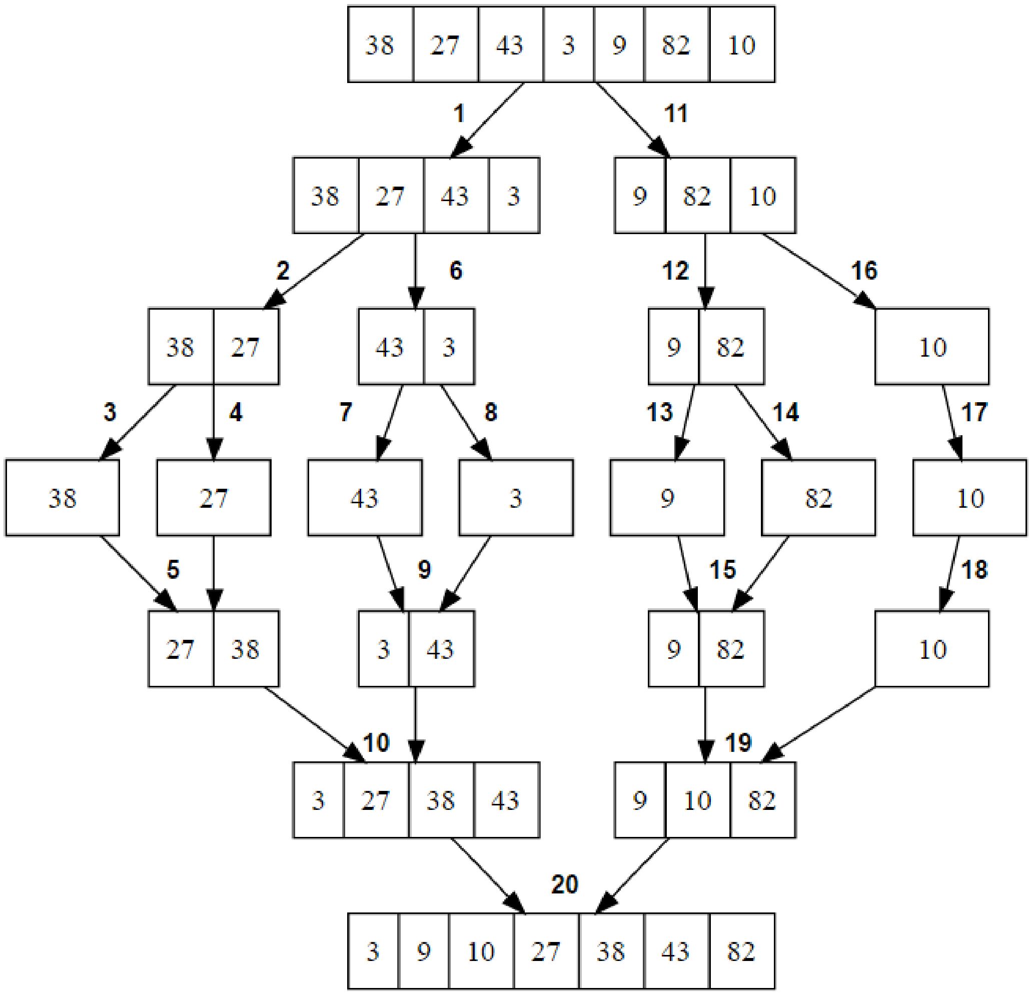
How Merge sort works?

Merge sort is a divide and conquer algorithm. Like all divide and conquer algorithms, merge sort divides a large array into two smaller subarrays and then recursively sort the subarrays. Basically, there are two steps are involved in whole process –

1. Divide the unsorted array into n subarrays, each of size 1 (an array of size 1 is considered sorted).
2. Repeatedly merge subarrays to produce new sorted subarrays until only 1 subarray is left which would be our sorted array.

Below diagram shows top-down view of recursive merge sort algorithm used to sort an array of 7 integers.



Merge Sort Performance:

Worst case time complexity of merge sort is `O(nlog(n))`. The recurrence relation is

`T(n) = 2T(n/2) + cn = O(nlog(n))`

The recurrence basically summaries merge sort algorithm – Sort two lists of half the size of the original list, and add the n steps taken to merge the resulting two lists.

Auxiliary space required by it is `O(n)`.