UNIVERSITY INSTITUTE OF COMPUTING

MASTER OF COMPUTER APPLICATIONS

DESIGN AND ANALYSIS OF ALGORITHMS

24CAT-611

UNIT-2

DISCOVER . LEARN . EMPOWER

Topics to be covered

- Divide and Conquer Method
- Binary Search
- Merge Sort
- Quick Sort

Merge Sort

- Merge Sort is a Divide and Conquer algorithm. It divides input array in two halves, calls itself for the two halves and then merges the two sorted halves. **The merge() function** is used for merging two halves. The merge(arr, l, m, r) is key process that assumes that arr[l..m] and arr[m+1..r] are sorted and merges the two sorted sub-arrays into one. See following C implementation for details.

How Merge Sort Works?

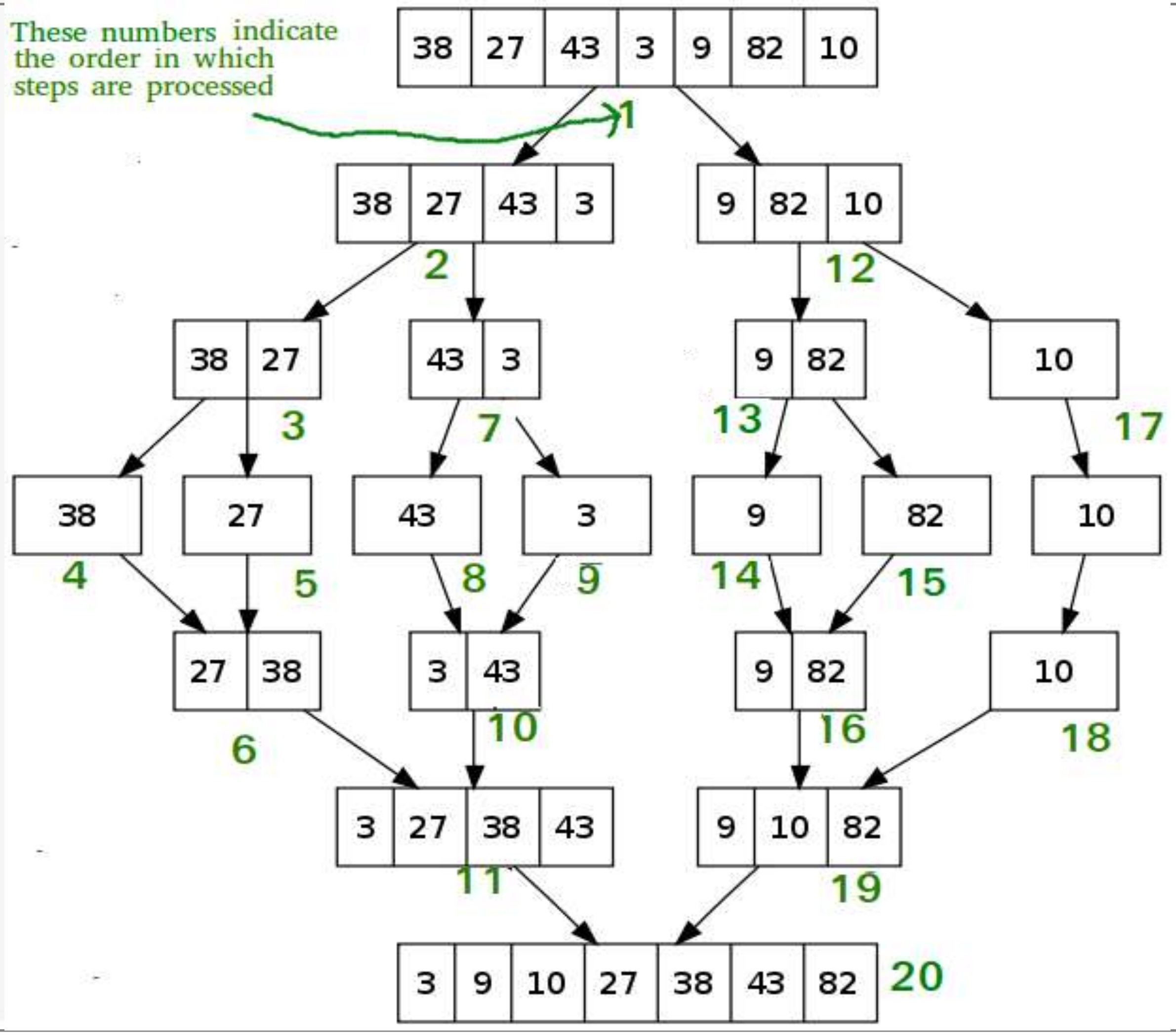


Figure1: Example of Merge sort

Contd…

- Merge sort is a sorting technique based on divide and conquer technique. With worst-case time complexity being O(n log n), it is one of the most respected algorithms.
- Merge sort first divides the array into equal halves and then combines them in a sorted manner.

Contd…

To understand merge sort, we take an unsorted array as the following –

We know that merge sort first divides the whole array iteratively into equal halves unless the atomic values are achieved. We see here that an array of 8 items is divided into two arrays of size 4.

Contd…

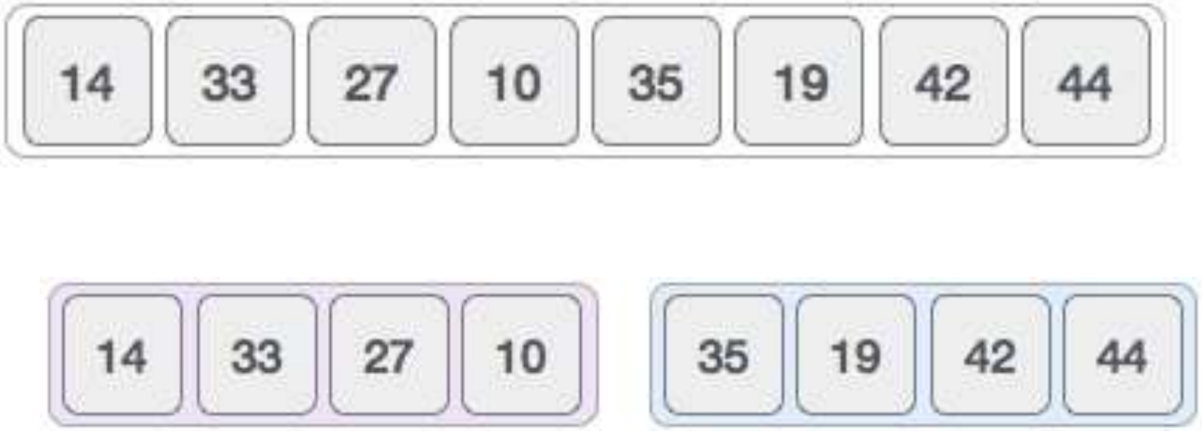


Figure2: Example of Merge sort

Contd…

- This does not change the sequence of appearance of items in the original. Now we divide these two arrays into halves.

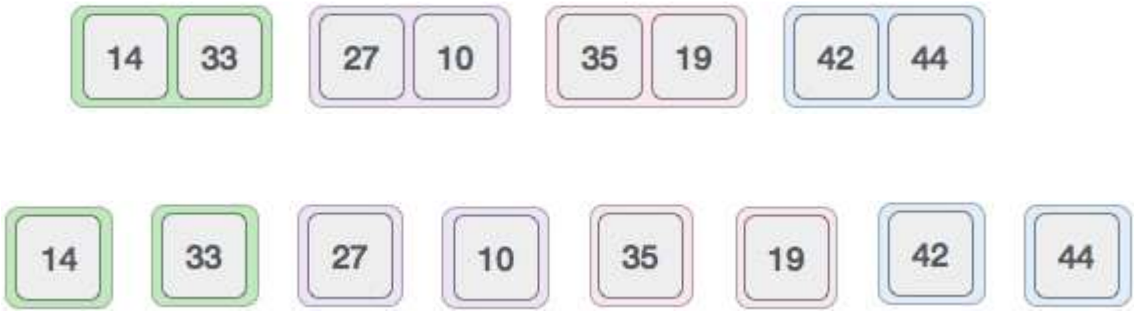


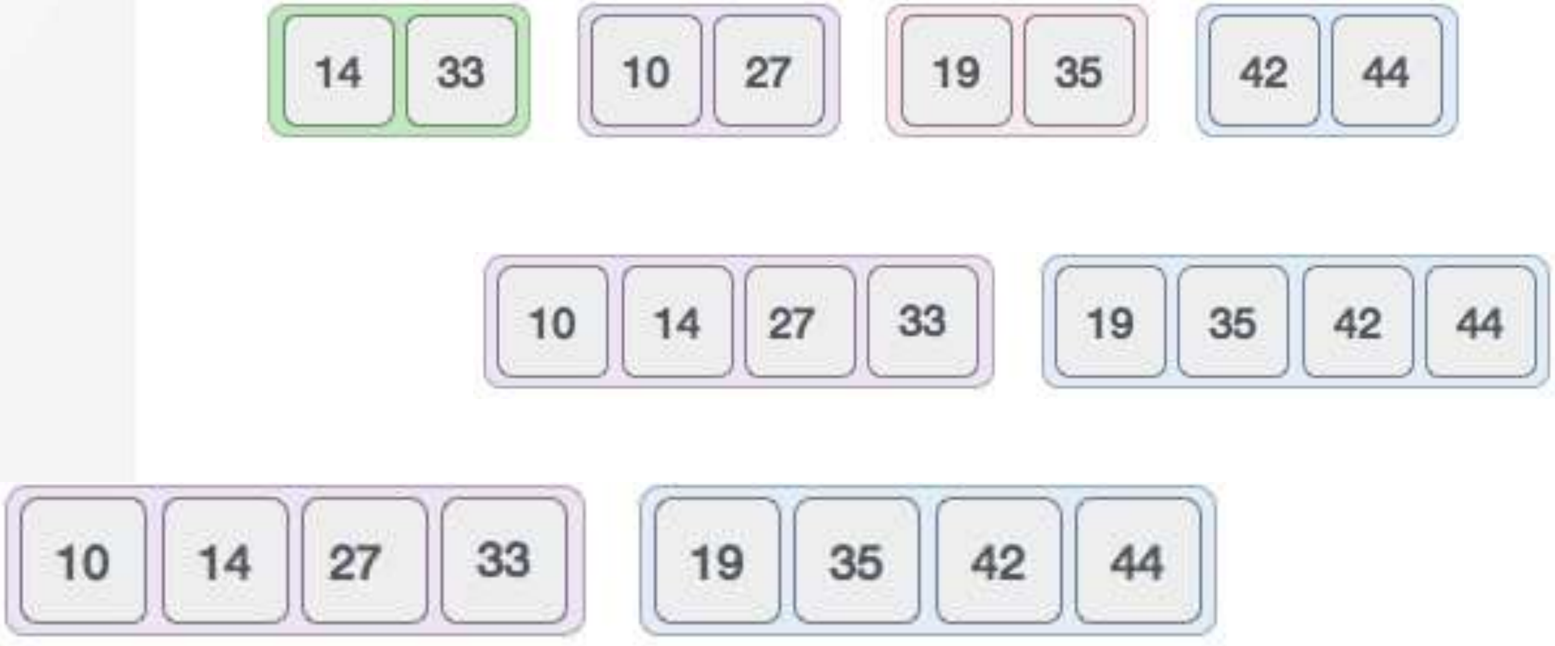
Figure3: Merge sort

We further divide these arrays and we achieve atomic value which can no more be divided.

Contd…

- Now, we combine them in exactly the same manner as they were broken down. Please note the color codes given to these lists.
- We first compare the element for each list and then combine them into another list in a sorted manner. We see that 14 and 33 are in sorted positions. We compare 27 and 10 and in the target list of 2 values we put 10 first, followed by 27. We change the order of 19 and 35 whereas 42 and 44 are placed sequentially.

Contd…



14 33 | 10 27 | 19 35 | 42 44

10 14 27 33 | 19 35 42 44

10 14 27 33 | 19 35 42 44

Figure 4: Swapping and sorting

Divide, Conquer and Combine

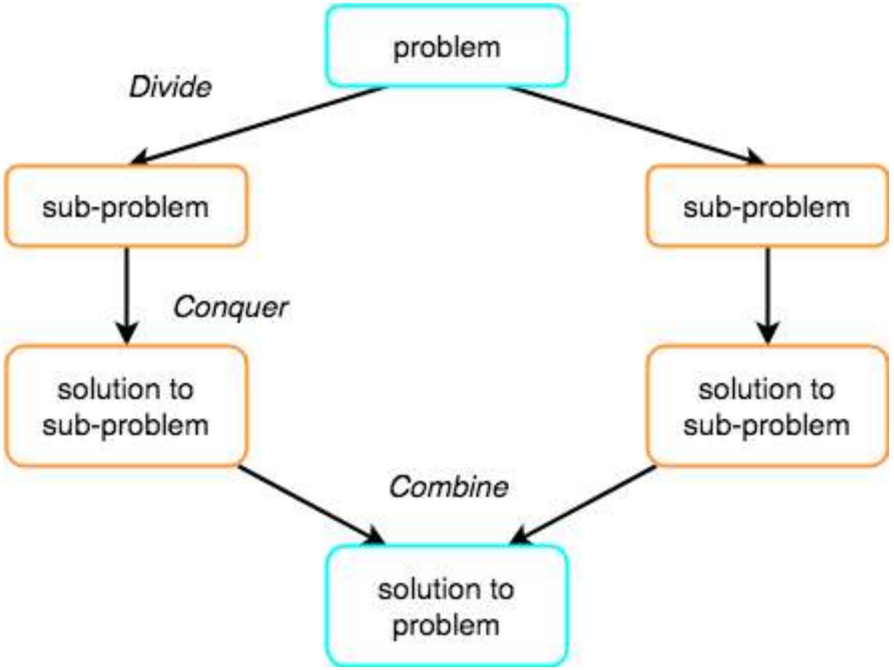


Figure 5: Divide, conquer and Combine

Complexity Analysis

- Worst Case Time Complexity [Big-O]: **O(n*log n)**
- Best Case Time Complexity [Big-omega]: **O(n*log n)**
- Average Time Complexity [Big-theta]: **O(n*log n)**
- Space Complexity: **O(n)**

Figure 5: Divide, conquer and Combine

Frequently asked questions

- What do you understand by merge sort?
- Define complexity of merge sort.

References

[1] https://www.tutorialspoint.com/data_structures_algorithms/images/unsorted_array.jpg

[2] https://www.tutorialspoint.com/data_structures_algorithms/images/merge_sort_divide_1.jpg

[3] https://www.tutorialspoint.com/data_structures_algorithms/images/merge_sort_combine_1.jpg

[4] https://www.tutorialspoint.com/data_structures_algorithms/images/merge_sort_combine_1.jpg

[5] https://www.studytonight.com/data-structures/images/divide-conquer.png

Books Reference

1. Introduction to Algorithms by Coreman, Leiserson, Rivest, Stein.
2. Fundamentals of Algorithms by Ellis Horwitz, Sartaj Sahni, Sanguthevar Rajasekaran

NAAC GRADE A+

THANK YOU