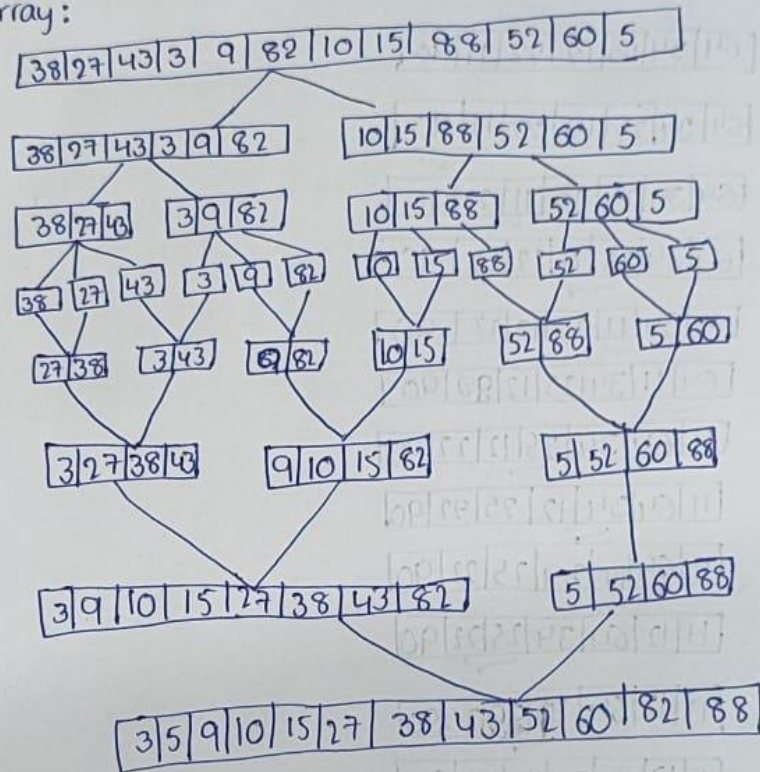


1. Sort the array using merge sort data and longer stage by
 [38, 27, 43, 3, 9, 10, 15, 88, 52, 60, 5] using analyze complexity of algorithm

sol: given array:

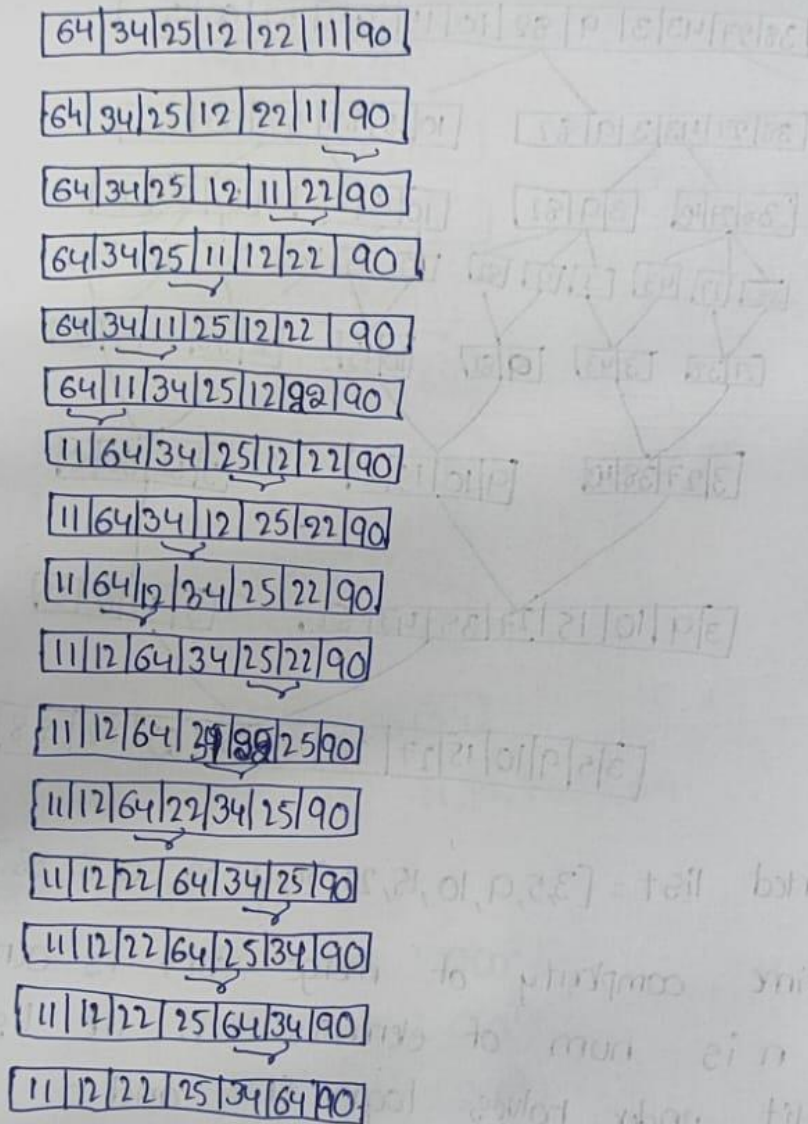


\therefore sorted list = [3, 5, 9, 10, 15, 27, 38, 43, 52, 60, 82, 88]

The time complexity of merge sort is $O(n \log n)$
 where n is num of elements in the list
 is split under halves $\log n$ time, and n
 Merging is all the elements at each level $O(n)$ time.

2. Sort the array 64, 34, 25, 12, 22, 90 using bubble. What is the time complexity of selection sort in the list worst and average case.

Sol:



3. Sort the array 64, 25, 12, 22, 11 using selection sort. What is the T.C. of selection sort in the best, worst and average class.

sol

64 25 12 22 11

25 64 12 22 11

25 12 64 22 11

25 12 22 64 11

25 12 22 11 64

12 25 22 11 64

12 22 25 11 64

12 22 11 25 64

12 11 22 25 64

11 12 22 25 64

∴ Sorted list is 11, 12, 22, 25, 64.

Time complexity:-

Best case : $O(n^2)$

Average case : $O(n^2)$

Worst case : $O(n^2)$

4. Given an array of $[4, -2, -5, 3, 10, -5, 2, 3, -3, 6, 7, -4, 1, 9, -1, 0, -6, -3, 11, 9]$ integers sort the following elements using analyze time complexity.

Sol: Insert 4, -2

$[-2, 4]$

Insert 5

$[-2, 4, 5]$

Insert 10

$[-2, 4, 5, 10]$

Insert -5

$[-5, -2, 3, 4, 5, 10]$

Insert 2

$[-5, -2, 2, 3, 4, 5, 10]$

Insert -3

$[-5, -3, -2, 2, 3, 4, 5, 10]$

Insert 6

$[-5, -3, -2, 2, 3, 4, 5, 6, 8, 10]$

Insert 7

$[-5, -3, -2, 2, 3, 4, 5, 6, 7, 8, 10]$

Insert -4

$[-5, -4, -3, -2, 2, 3, 4, 5, 6, 7, 8, 10]$

Insert 9

$[-5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

Insert -1

$[-5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

Insert -6

$[-6, -5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

Insert -9

$[-9, -6, -5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

Time complexity:-

Best case :- $O(n)$. This occurs when the array is already sorted the inner loop runs zero times for every element.

Average case :- This happens because on an array the algorithm will have to move half of the elements for each insertion.

Worst case: $O(n^2)$. This happens when the array is sorted in reverse order each insertion takes $O(n)$ times

5. Sort the following elements using insertion sort using brute force approach strategy [38, 27, 43, 3, 9, 82, 10, 15, 33, 52, 60, 5] and analyze complexity of the algorithm.

Insert 38, 27

[27 | 38]

43:-

[27 | 38 | 43]

3:-

[3 | 27 | 38 | 43]

9:-

[3 | 9 | 27 | 38 | 43]

82:-

[3 | 9 | 27 | 38 | 43 | 82]

10:-

[3 | 9 | 10 | 27 | 38 | 43 | 82]

15:-

[3 | 9 | 10 | 15 | 27 | 38 | 43 | 82]

33:-

[3 | 9 | 10 | 15 | 27 | 33 | 38 | 43 | 82]

52:-

[3 | 9 | 10 | 15 | 27 | 33 | 38 | 43 | 52 | 82]

60:-

[3 | 9 | 10 | 15 | 27 | 33 | 38 | 43 | 52 | 60 | 82]

5:-

[3 | 5 | 9 | 10 | 15 | 27 | 33 | 38 | 43 | 52 | 60 | 82]

Time complexity:-

Base case: $O(n)$

Average case: $O(n^2)$

Worst case: $O(n^2)$