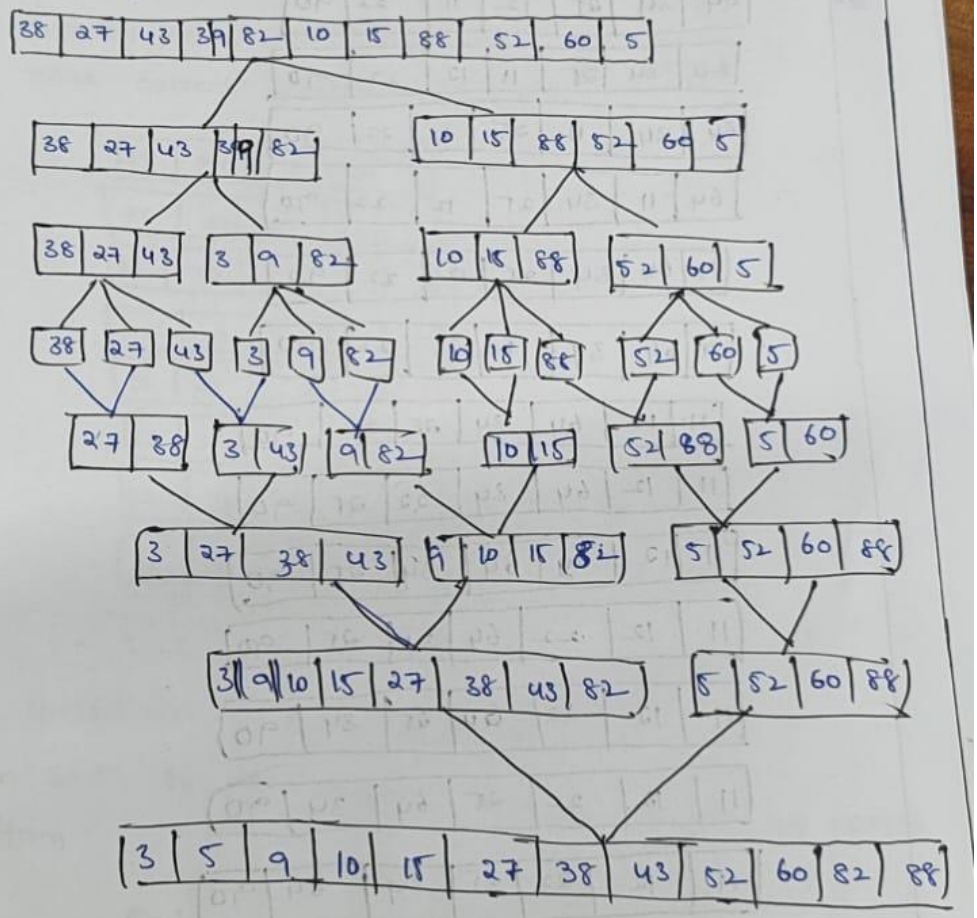
① Sort the array using merge sort data and tracking language stage by [38, 27, 43, 39, 82, 10, 15, 88, 52, 60, 5] and analyze the time complexity of the algorithm

Sol: Given array: merge sort



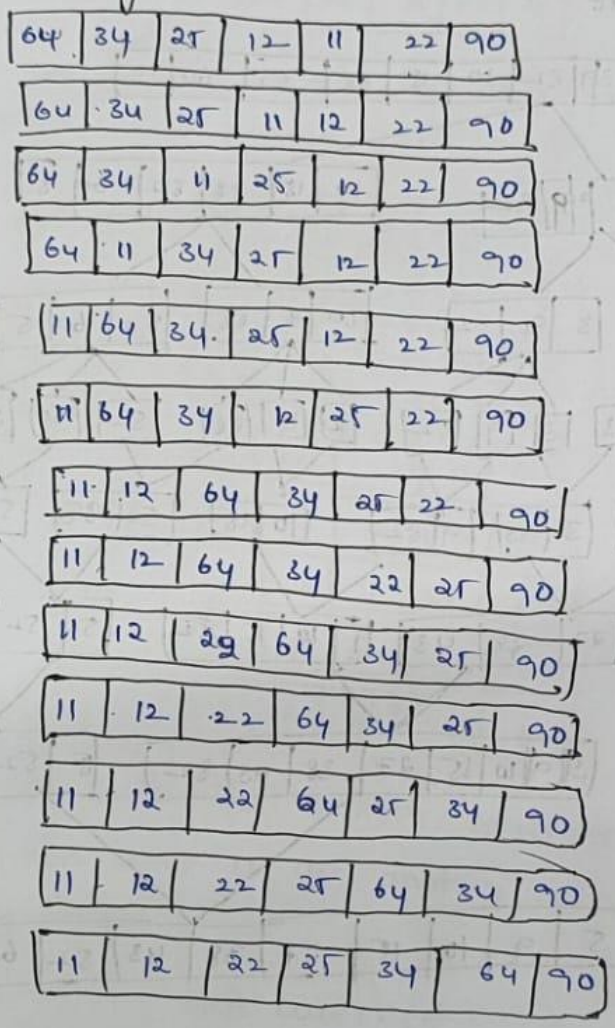
∴ sorted array = [3, 5, 9, 10, 15, 27, 38, 43, 52, 60, 82, 88]

The time complexity of merge sort is O(n log n) where n is the no. of elements in the list is split into halves log n times and n

merging is all the elements at each takes O(n) times

(12) sort the array 64, 34, 25, 12, 22, 11, 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 case?

64	25	12	22	11

In the selection we will fix that from the largest element in the there correct partition first so

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

The sorted list is 11, 19, 22, 25, 64.

Selection sort is an another simple comparison sorted algorithm

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, 8, -3, 6, 7, -4, 1, 9, -1, 0, -6, -8, 11, -9] integers, sort the following elements using insertion sort using Brute Force Approach strategy analyze complexity of the algorithm.

Sol: Given array

[4, -2, 5, 3, 10, -5, 2, 8, -3, 6, 7, -4, 1, 9, -1, 0, -6, -8, 11, -9]

Insert 4, -2

-2	4

Insert 5

-2	4	5

Insert 10.

-2	3	4	8	10

Insert -5

-5	-2	3	4	8	10

Insert 2

-5	-2	2	3	4	8	10

Insert 8

-5	-2	2	3	4	5	8	10

Insert -3

-5	-3	-2	2	3	4	5	8	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	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 -8

-8	-6	-5	-4	-3	-2	-1	1	2	3	4	5	6	7	8	9	10

Insert -9

-9	-8	-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 3vo times for every element

Average case.

This happens because an average the algorithm will have to more half of the element for each insertion

Worst case: $O(n^2)$

This occurs 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, 38, 10, 15, 34, 52, 15, 60, 5) and analyse complexity of the algorithm

Sol: Insert 38, 27.

27	38

Insert 43:

27	38	43

Insert 3:

3	27	38	43

Insert 82:

3	9	27	38	43	82

Insert 10:-

3	9	10	27	38	43	82

Insert 15:

3	9	10	15	27	38	43	82

Insert 88:

3	9	10	15	27	38	43	82	88

Insert 62 :

| 3 | 9 | 10 | 25 | 27 | 38 | 43 | 57 | 62 | 88 |

Insert 60 :

| 3 | 9 | 10 | 25 | 27 | 38 | 43 | 52 | 60 | 82 | 88 |

Insert 5

| 3 | 5 | 9 | 10 | 25 | 27 | 38 | 43 | 52 | 60 | 82 | 88 |

Time complexity :

Best case : $O(n)$

Average case : $O(n^2)$

Worst case : $O(n^2)$