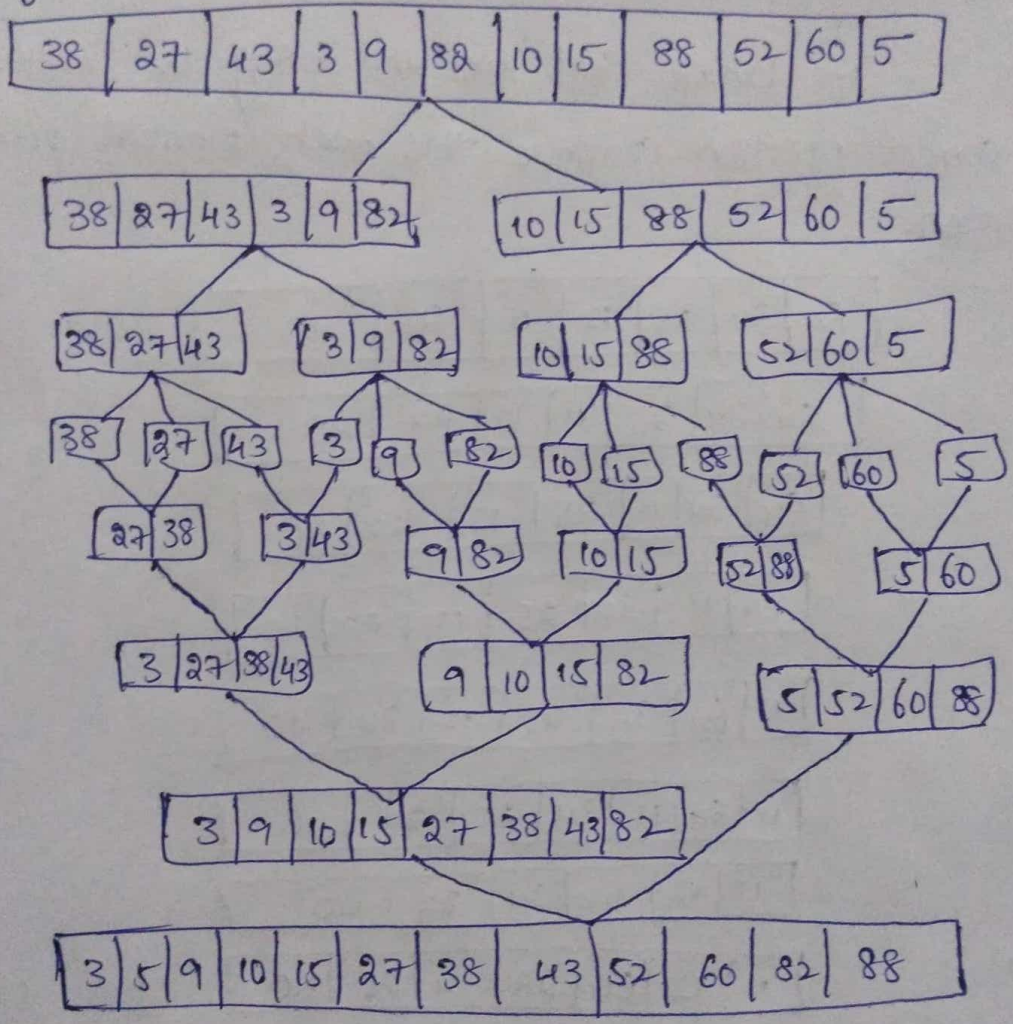
① Sort the following elements using merge sort divide and conquered [38, 27, 43, 3, 9, 87, 10, 15, 88, 52, 60, 5] using and analysize time complexity of algorithm.

Given array



∴ Sorted list = (3, 5, 9, 10, 15, 27, 38, 43, 52, 60, 82, 88)

Time Complexity :=

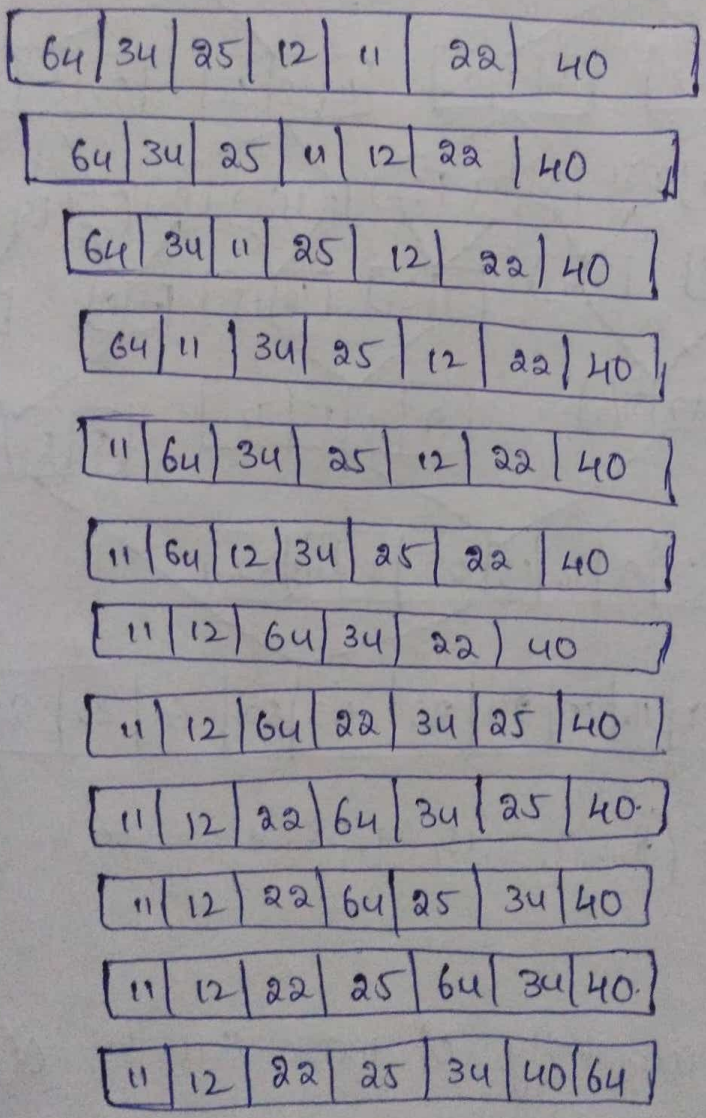
Time complexity of merge sort is O(nlogn)

'n' is the num of elements in the list is O(nlogn)

(2) Sort the array 64, 34, 25, 12, 11, 90 using bubble sort what is the time complexity of solution and sort in best, average, worst cases.

Given array = 64, 34, 25, 12, 22, 11, 90.

In bubble sort we will bring the smallest element in correct position continue this each element reach it's position.



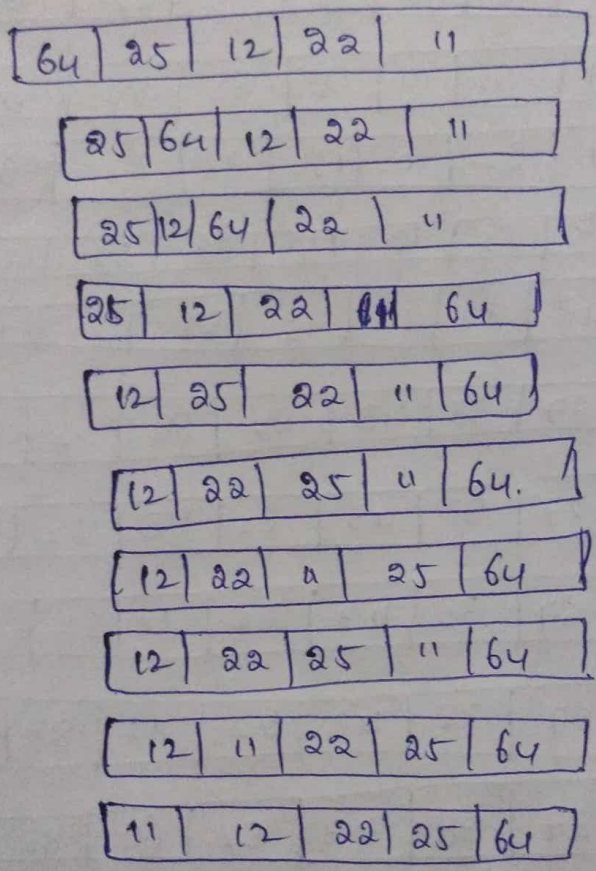
Time complexity:- Best case :- $O(n^2)$ worst case :- $O(n^2)$

Average case :- $O(n^2)$

(3) Sort the array 64, 25, 12, 22, 11 using Selection Sort. what is the time complexity of selection sort in the best, worst and average cases.

Given array :- 64, 25, 12, 22, 11

In the Selection Sort we will write from the largest element in those correct position best so.



∴ The Sorted list is 11, 12, 22, 25, 64.

Time complexity :- Selection Sort

Best case :- $O(n^2)$

Average case :- $O(n^2)$

worst case :- $O(n^2)$

∴ The selection sort has a time complexity $O(n^2)$ if always through same no.of comparisons.

(4) Sort the following elements using insertion Sort using Brute force approach Strategy [38, 127, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5] and analyze complexity of algorithm.

Given array

[38, 127, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5]

38	27	43	3	9	82	10	15	88	52	60	5
27	38	43	3	9	82	10	15	88	52	60	5
27	38	43	3	9	82	10	15	88	52	60	5
3	27	38	43	9	82	10	15	88	52	60	5
3	9	27	38	43	82	10	15	88	52	60	5
3	9	10	15	27	38	43	82	88	52	60	5
3	9	10	15	27	38	43	82	88	52	60	5
3	9	10	15	27	38	43	82	88	52	60	5
3	9	10	15	27	38	43	52	82	88	60	5
3	9	10	15	27	38	43	52	60	82	88	5
3	5	9	10	15	27	38	43	52	60	82	88

Time complexity :=

Best case := $O(n^2)$

Average case := $O(n^2)$

worst case := $O(n^2)$

(5) Given 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 approch strategy and analysize time complexity of algorithm.

Insert '4' = [4]

Insert '-2' = [-2, 4]

Insert '5' = [-2, 4, 5]

Insert '3' = [-2, 4, 3, 5]

Insert '10' = [-2, 3, 4, 5, 10]

Insert '-5' = [-5, -2, 3, 4, 5, 10]

Insert '2' = [-5, -2, 2, 3, 4, 5, 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 '1' = [-5, -4, -3, -2, 1, 2, 3, 4, 5, 6, 7, 8, 10]

Insert '9' = [-5, -4, -3, -2, 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 '0' = [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Insert '-6' = [-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Insert '-8' = [-8, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

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

Time complexity :=

Best case := $O(n)$

Average case := $O(n^2)$

worst case := $O(n^2)$