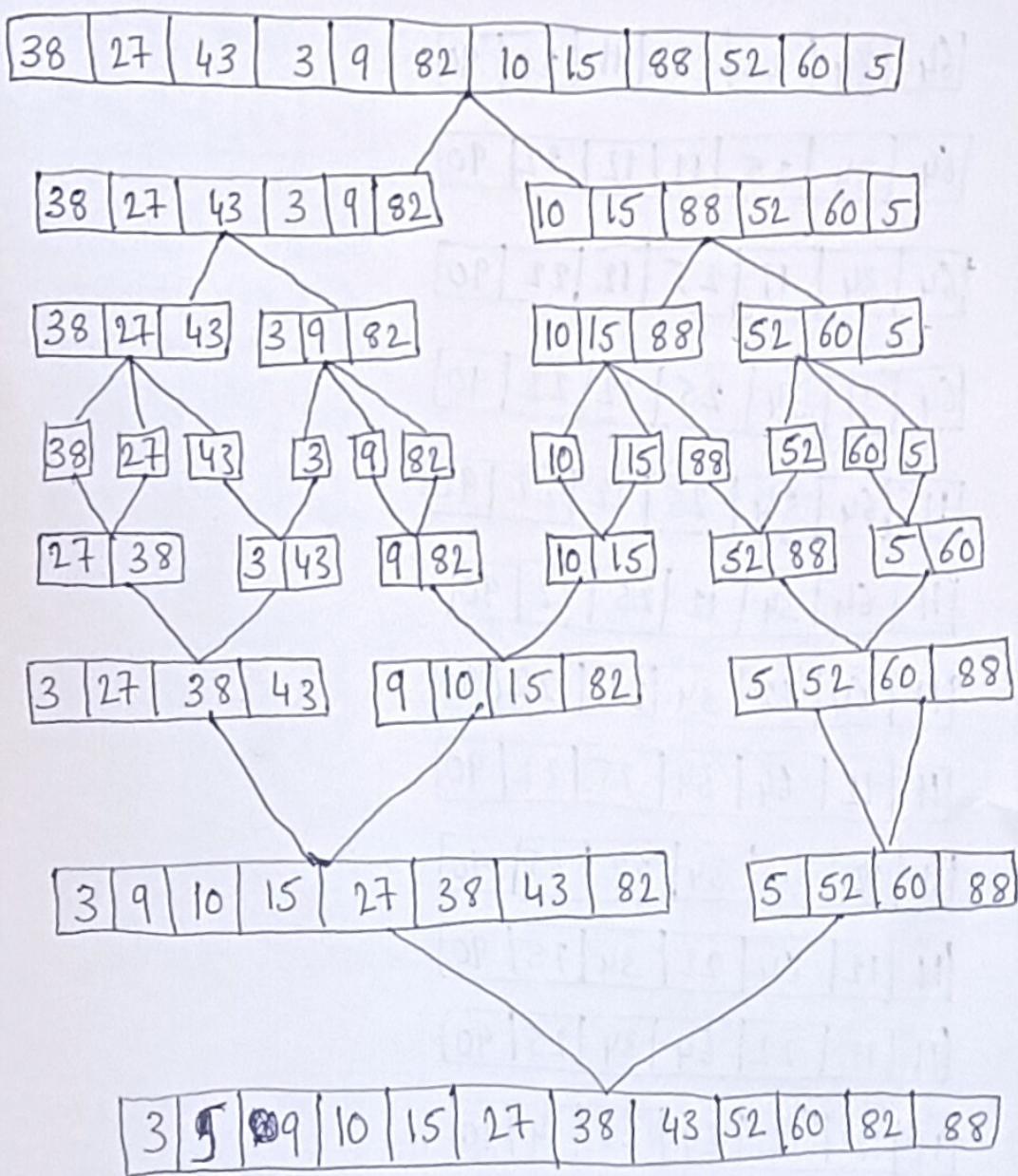


1) Sort the array using merge sort data and lengths stage by [38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5] using and analyze complexity of algorithm.

Sol: Given array : merge sort



∴ 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 the num of elements in the list is split into halves  $\log n$  time and  $n$ .

Merging is all the elements at each level takes  $O(n)$  time.

3) 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:

64	34	25	12	22	11	90
----	----	----	----	----	----	----

64	34	25	12	11	22	90
----	----	----	----	----	----	----

64	34	25	11	12	22	90
----	----	----	----	----	----	----

64	34	11	25	12	22	90
----	----	----	----	----	----	----

64	11	34	25	12	22	90
----	----	----	----	----	----	----

11	64	34	25	12	22	90
----	----	----	----	----	----	----

11	64	34	12	25	22	90
----	----	----	----	----	----	----

11	12	64	34	25	22	90
----	----	----	----	----	----	----

11	12	64	34	22	25	90
----	----	----	----	----	----	----

11	12	64	22	34	25	90
----	----	----	----	----	----	----

11	12	22	64	34	25	90
----	----	----	----	----	----	----

11	12	22	25	64	34	90
----	----	----	----	----	----	----

11	12	22	25	34	64	90
----	----	----	----	----	----	----

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 cases?

64	25	12	22	11
----	----	----	----	----

In the selection we will fix that from the longest elements in these correct position 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, 12, 22, 25, 64.

Time complexity:

selection sort is an another simple comparison sorted algorithm.

Best case:  $O(n^2)$

Average case:  $O(n^2)$

Worst case:  $O(n^2)$

Given an array of  $\{4, -2, -5, 3, 10, -5, 2, 3, 6, 7, -4, 1, 9, -1, 0, -6, -3\}$  integers sort the following elements using analyse time complexity.

Given array:

$4, -2, -5, 3, 10, -5, 2, 3, -3, 6, 7, -4, 1, 9, -1, 0, -6, -3, 11, -9$ .

Insert 4, -2

-2	4
----	---

Insert 5

-2	4	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 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 -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	-6	-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 average the algorithm will have to move half of the elements for each insertion.

### Worst case: $O(n^2)$

This occurs when the array is sorted in reverse order each insertion takes  $O(n)$  times.

- Q) 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 analyse complexity of the algorithm.

Insert 38, 27.

27	38
----	----

Insert 43

27	38	43
----	----	----

Insert 3:

3	27	38	43
---	----	----	----

Insert 9:

3	9	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 33:

3	9	10	15	27	33	38	43	82
---	---	----	----	----	----	----	----	----

Insert 52:

3	9	10	15	27	33	38	43	52	82
---	---	----	----	----	----	----	----	----	----

Insert 60:

3	9	10	15	27	33	38	43	52	60	82
---	---	----	----	----	----	----	----	----	----	----

Insert 5:

3	5	9	10	15	27	33	38	43	52	60	82
---	---	---	----	----	----	----	----	----	----	----	----

Time complexity:

Best case:  $O(n)$

Average case:  $O(n^2)$

Worst case:  $O(n^2)$