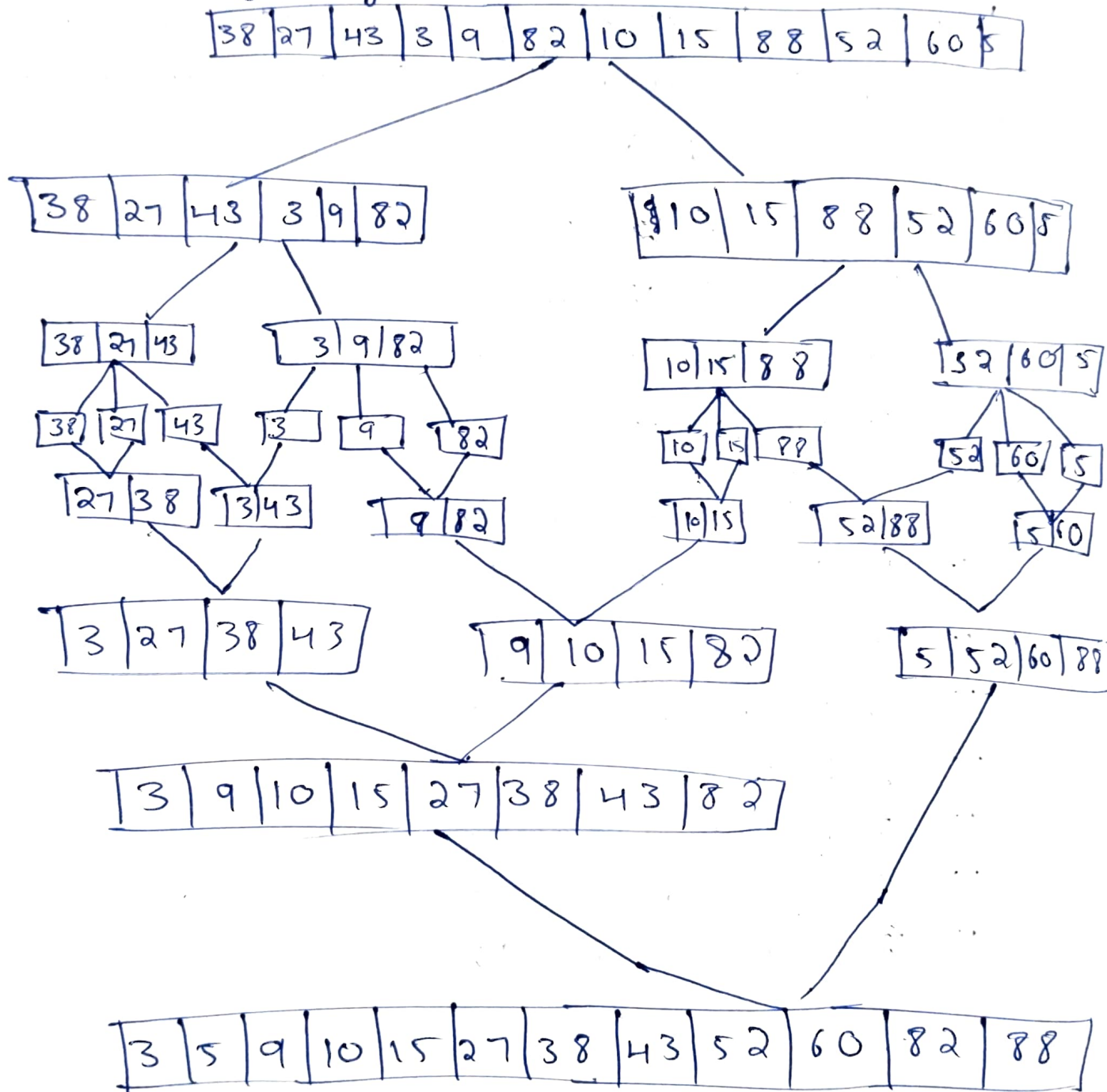


1. Sort the following elements using merge sort divide and conquer stage by [38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5] using analyze time complexity.

A. Given array :- merge sort



Time complexity! - Time complexity of merge sort is $O(n \log n)$ where n is the num of elements in the list This is because the list is split into halves $\log n$ times and n .

2. Sort the array 64, 34, 25, 12, 22, 11, 90 using bubble sort what is the time complexity of selection sort in the best, worst and average cases.

A. Given array:

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

In bubble sort we bring from smallest element in their correct position continue this until each element reach there in correct position.

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	12	34	25	22	90
11	12	64	34	25	22	90
11	12	34	64	25	22	90
11	12	64	22	34	25	90
11	12	22	64	25	34	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 time complexity of selection sort in the best, worst and average cases?

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

A. In the selection sort we will fix that from the largest element in their 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
----	----	----	----	----

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

The sorted list is 11, 12, 22, 25, 64

TIME COMPLEXITY:- selection sort is 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, 1) insert 4, -2

Insert 4, -2

-2	4
----	---

Insert 5

-2	4	5
----	---	---

Insert 3

-2	3	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 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, 0, 6, -8, 1

-8	-6	-5	-4	-3	-2	0	1	2	3	4	5	6	7	8	9	10
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---	----

Time complexity:-

Best case! $O(n)$ This occurs when the array is already sorted.

Average case! $-O(n^2)$ This happens because on average algorithm will have to move 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, 82, 10, 15, 88, 52, 60, 5) and analyze 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, 15

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

Insert 88

3	9	10	15	27	38	43	82	88
---	---	----	----	----	----	----	----	----

Insert 52

3	9	10	15	27	38	43	52	82	88
---	---	----	----	----	----	----	----	----	----

Insert 60

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

Insert 5

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

Time Complexity:-

Best case :- $O(n)$

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

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