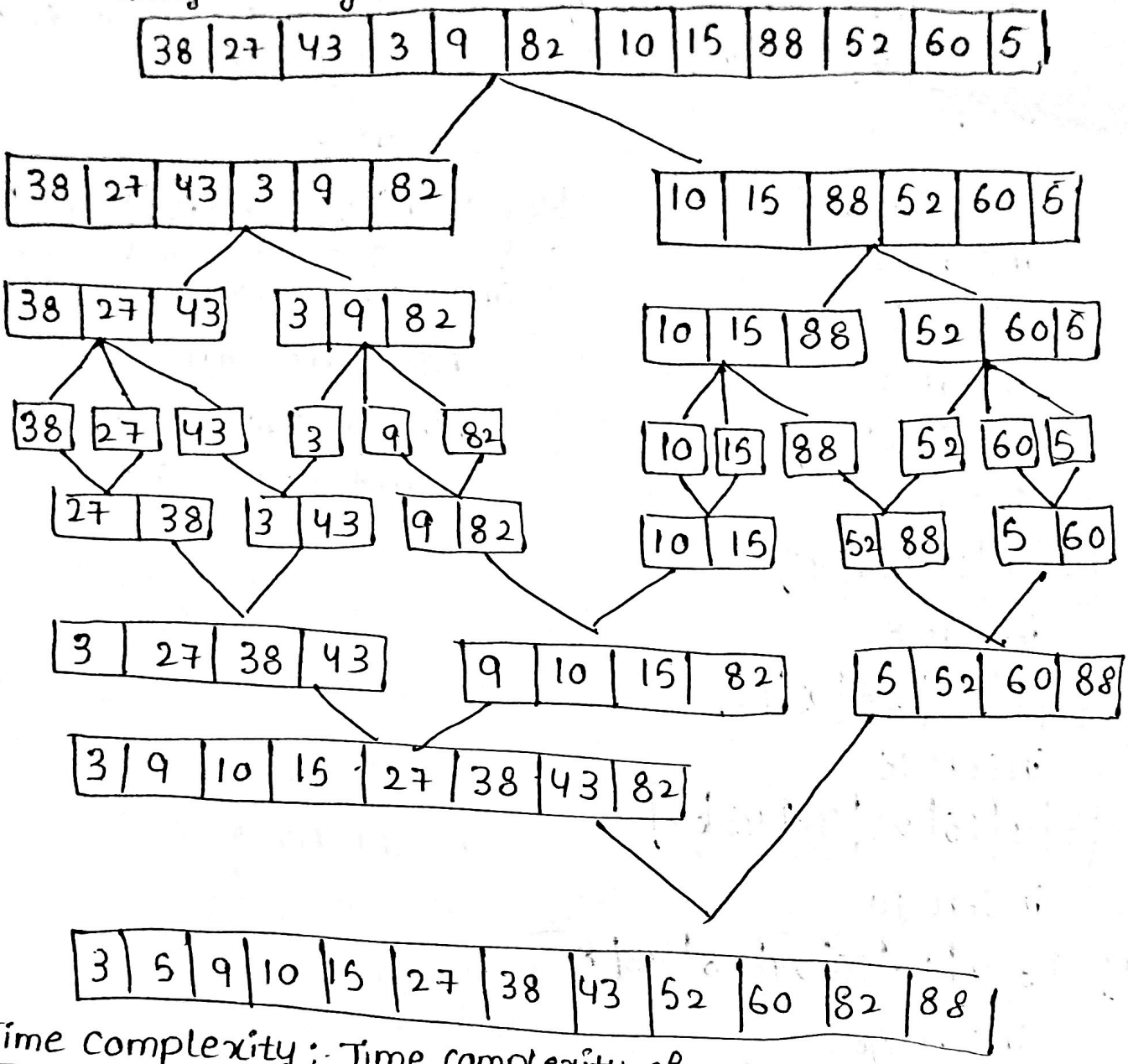
1. Sort the following elements using merge sort divide and Conquer stage by [38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5] using and time complexity.

A. Given array :- merge sort



Time complexity :- Time complexity of merge sort is O(nlogn) where n is the num of elements in the list this is because the list is split into halves logn 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 | 90 |

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

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 case?

64	25	12	22	11

in the selection sort we will fix that from the largest element in there 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	25	64
11	12	22	25	64

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

Time complexity: selection sort is an another simple comparision sorted algorithm.

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

Average case: $O(n^2)$

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

4. Give 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

-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 1

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

insert 9

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

insert 0

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

insert -6

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

insert -8

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

insert 11

-8	-6	-5	-3	-2	0	1	2	3	4	5	6	8	9	10	11

insert -9

-9	-8	-6	-5	-3	-2	0	1	2	3	4	5	6	8	9	10	11

Time complexity:

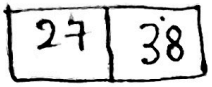
Best case:- $O(n)$. This occurs when the array is already sorted.

Average case $O(n^2)$: This happens because on average algorithm will have to move half of the element for each insertion.

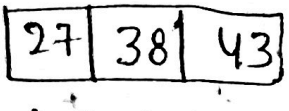
worst case : $O(n^2)$:- this occurs when the array is sorted in reverse order each insertion takes $O(n)$ times.

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.

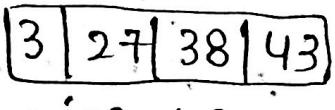
insert 38, 27



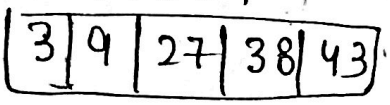
insert 43



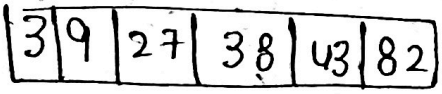
insert 3



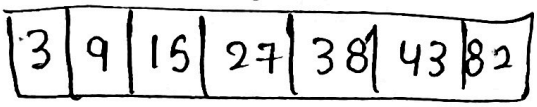
insert 9



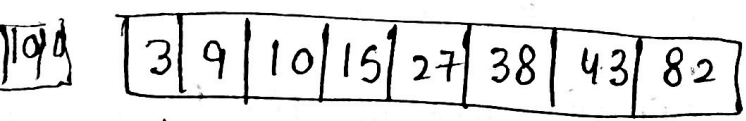
insert 82



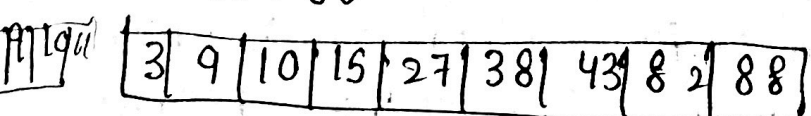
insert 15



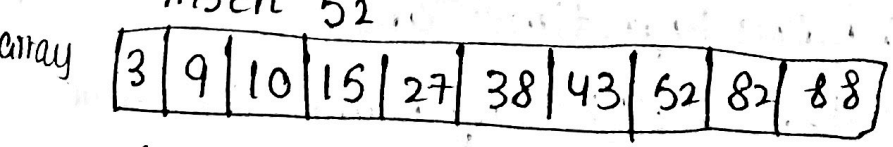
insert 10



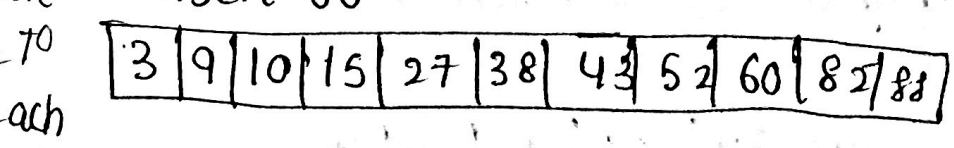
insert 88



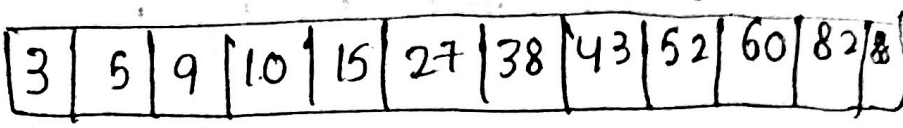
insert 52



insert 60



insert 5



Time complexity:

Best case = $O(n)$

Average case = $O(n^2)$

worst case = $O(n^2)$