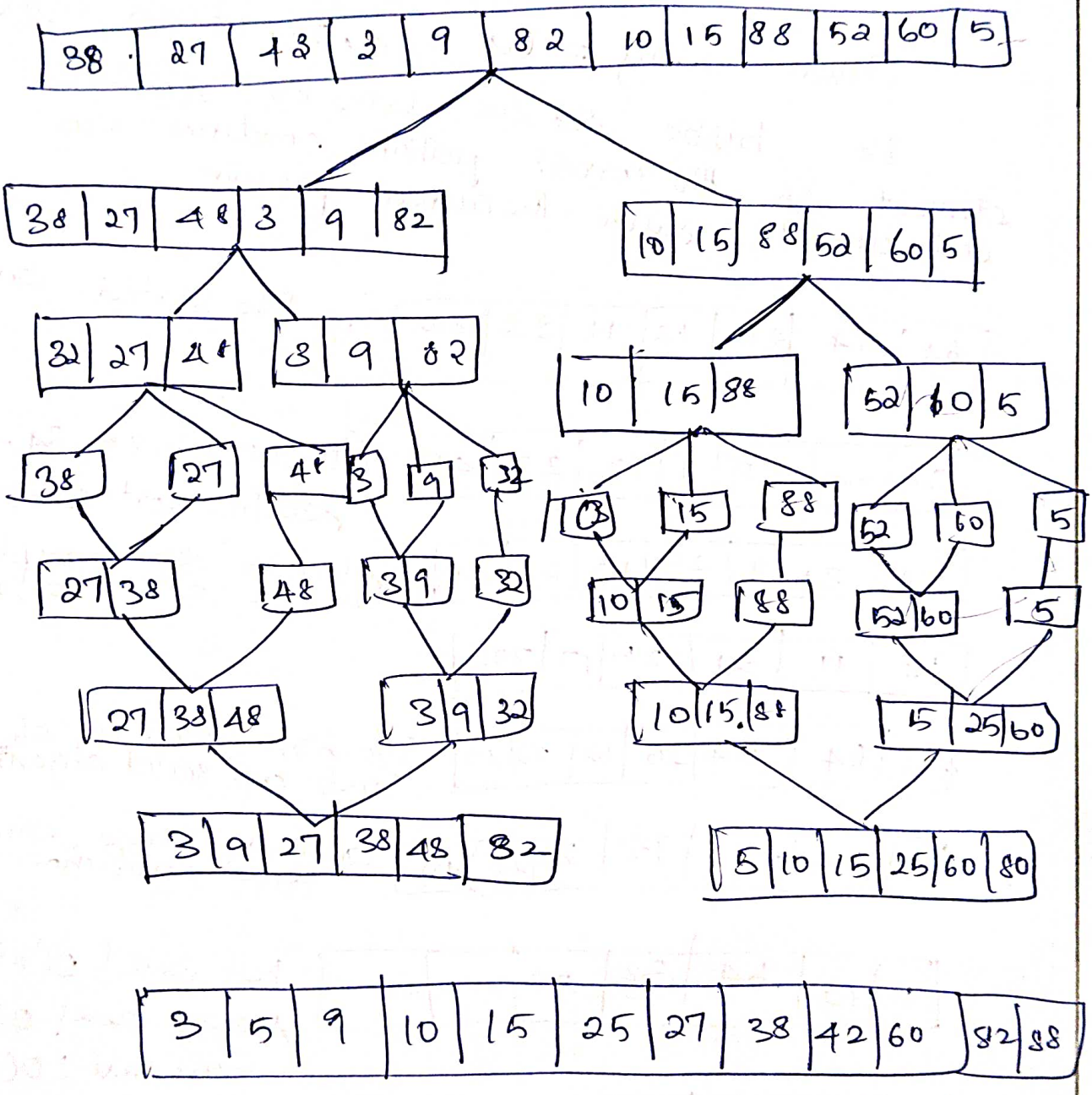
Austin Stejoh
192311455.

1) Sort the following Elements using merge sort divide and conquered [38, 27, 43, 3, 9, 82, 10, 15, 88, 52] using and analysing time complexity of the algorithm.

Soln:

Given array:



Sorted List = [3, 5, 9, 10, 15, 25, 27, 38, 43, 52, 60, 82, 88].

Time complexity: -

Time complexity of merge sort

is O(nlogn) his the num of elements in the list O(nlog s) increase log the input into values logn times and n Element of each time values O(n) times.

2- Sort the array 64, 34, 25, 121, 11, 99 using bubble Sort what is the time complexity of solution Sort in the best, worst, average case.

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

In bubble set eve bring the smallest element in the correct position continue this each element seach the current position.

64	34	25	12	11	22	40

∴ The sorted array.

64	34	25	11	12	22	40

(11, 12, 22, 25, 34, 40, 64)
selection sort complexity.

64	34	11	25	12	22	40

selection sort complexity
2 2 2 2 2 2 2 2.

64	11	34	25	12	22	40

11	64	34	25	12	22	40

Selection sort is an other single com sorted algorithm.

11	64	34	12	25	22	40

another single com sorted algorithm.

11	12	64	32	22	25	40

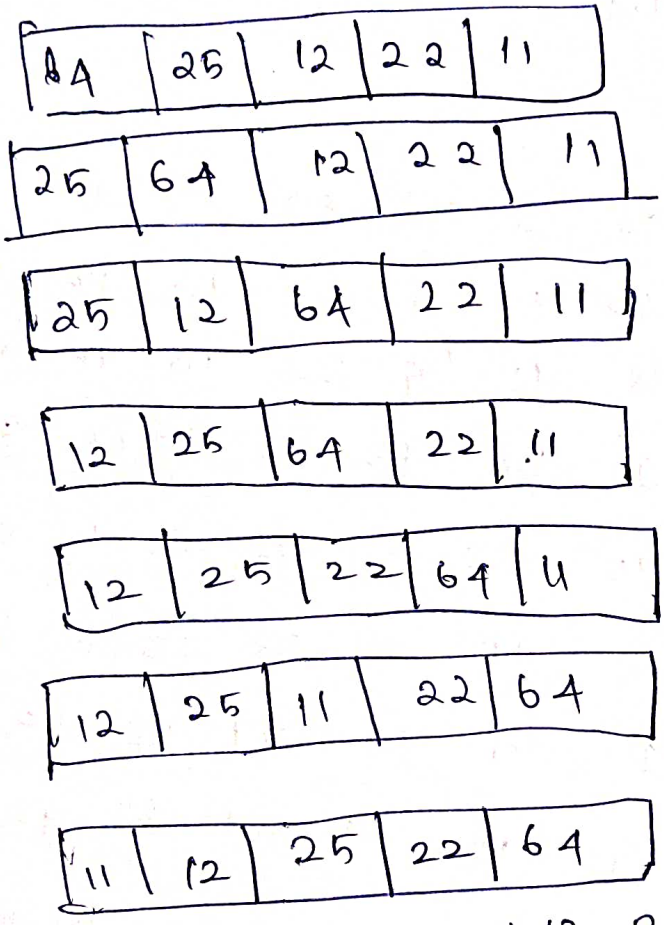
best case! $O(n^2)$.
Average case! $O(n^2)$
worst case: $O(n^2)$.

11	12	22	25	36	40	64

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.

given array:- 64, 25, 12, 22, 11.

In the selection are will the form the longest Element in the 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 $O(n^2)$ it always through some $O(n, o(n))$.

4) Sort the following elements using insertion sort using Brute Force Approach strategy. [38, 27, 43, 3, 9, 82, 10, 88, 56, 60, 5) and analyse complexity of the algorithm.

Soln:

Given Array:

[38, 27, 43, 3, 9, 32, 10, 15, 88, 52, 60, 5] -

Solve:

38 27 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 27 38 43 82 15 88 52 60 5

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

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

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

Time Complexity: Worst case: $O(n^2)$

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

Best case: $O(n)$

5. 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 element using insertion sort using Brute force Approach strategy analyse Complexity of algorithm.

Soln:

Insert: -4 = [4].

insert : --2 = [-2, 4].

insert : -5 = [-2, 4, 5].

Insert : -3 = [-2, 3, 4, 5].

insert : -10 = [-2, 3, 4, 10].

insert : --5 = [-2, -5, 3, 4, 10].

insert : 2 = [-2, -5, 2, 3, 4, 10].

insert : -8 = [-2, -5, 2, 3, 4, 8, 10].

insert : -3 = [-2, -3, -5, 2, 3, 4, 8, 10].

insert : --6 = [-2, -3, -5, 2, 3, 4, 6, 8, 10]

insert : -7 = [-2, -3, -5, 2, 3, 4, 6, 7, 8, 10].

insert : 1 = [-2, -3, -5, 1, 2, 3, 4, 6, 7, 8, 10]

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

Time Complexity: Best:- O(n). Average: $O(n^2)$.

Worst: $O(n^2)$.