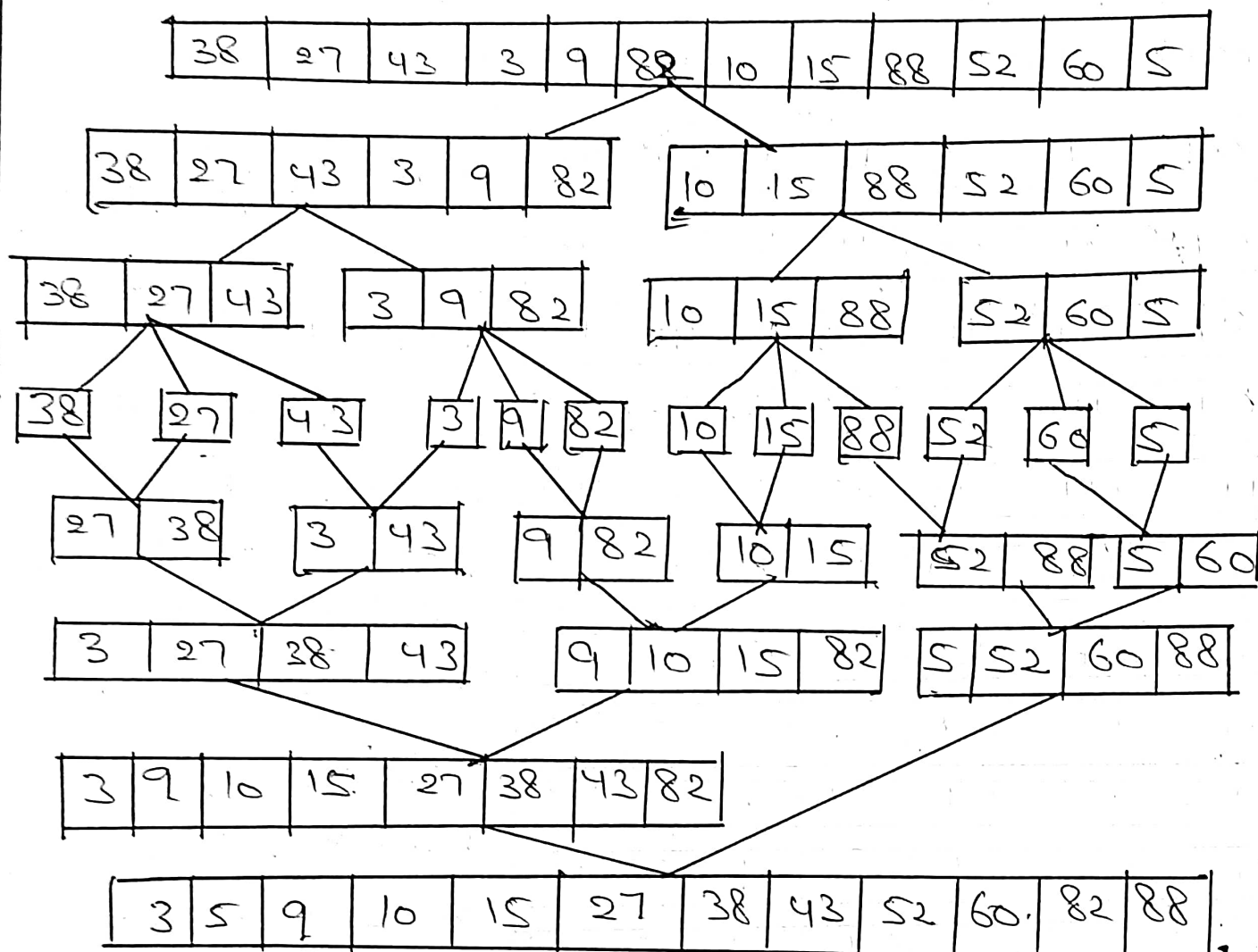


1) Sort the following elements using merge sort divide and conquer (38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5) using and analyze time complexity of the algorithm.

Sol, given array



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

Time complexity:

Time complexity of merge sort is $O(n \log n)$

n is the num. of elements in the list $O(n \log n)$ increase by the input into halves $\log n$ times and n element of each time takes $O(n)$ times

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

Sol/ Given array = 64 34 25 12 22 11 90

In bubble sort we bring the smallest element in the correct position continue this each element reach the current 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	34	12	25	22	90
----	----	----	----	----	----	----

11	64	12	34	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
----	----	----	----	----	----	----

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

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

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

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

∴ The sorted arrays

is

(11, 12, 22, 25, 34, 40, 64)

Selection Sort Complexity

Selection sort is an another single confusion sorted algorithm

best case : $O(n^2)$

Average : $O(n^2)$

worst case : $O(n^2)$

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

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.

sol) Given array : 64, 25, 12, 22, 11

In the selection sort we will find the largest element in the array and swap it with the element at the first position.

64	25	12	22	11
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 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 it always through same $O(n)$ comparisons.

Q) 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 the algorithm.

Sol) Given array

[38, 127, 43, 3, 9, 82, 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
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	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	9	10	15	27	38	43	82	88	52	60	5
3	9	10	15	27	38	43	52	88	88	60	5
3	9	10	15	27	38	43	52	60	82	88	5
3	5	9	10	27	38	43	52	60	82	88	5
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 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 Approach Strategy. Analyse complexity of algorithm

Sol) Insert: $-4 =$
 $[4]$

Insert: $-2 = [-2, 4]$

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: $-1 = [-5, -4, -3, -2, -1, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$

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

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: $O(n^2)$