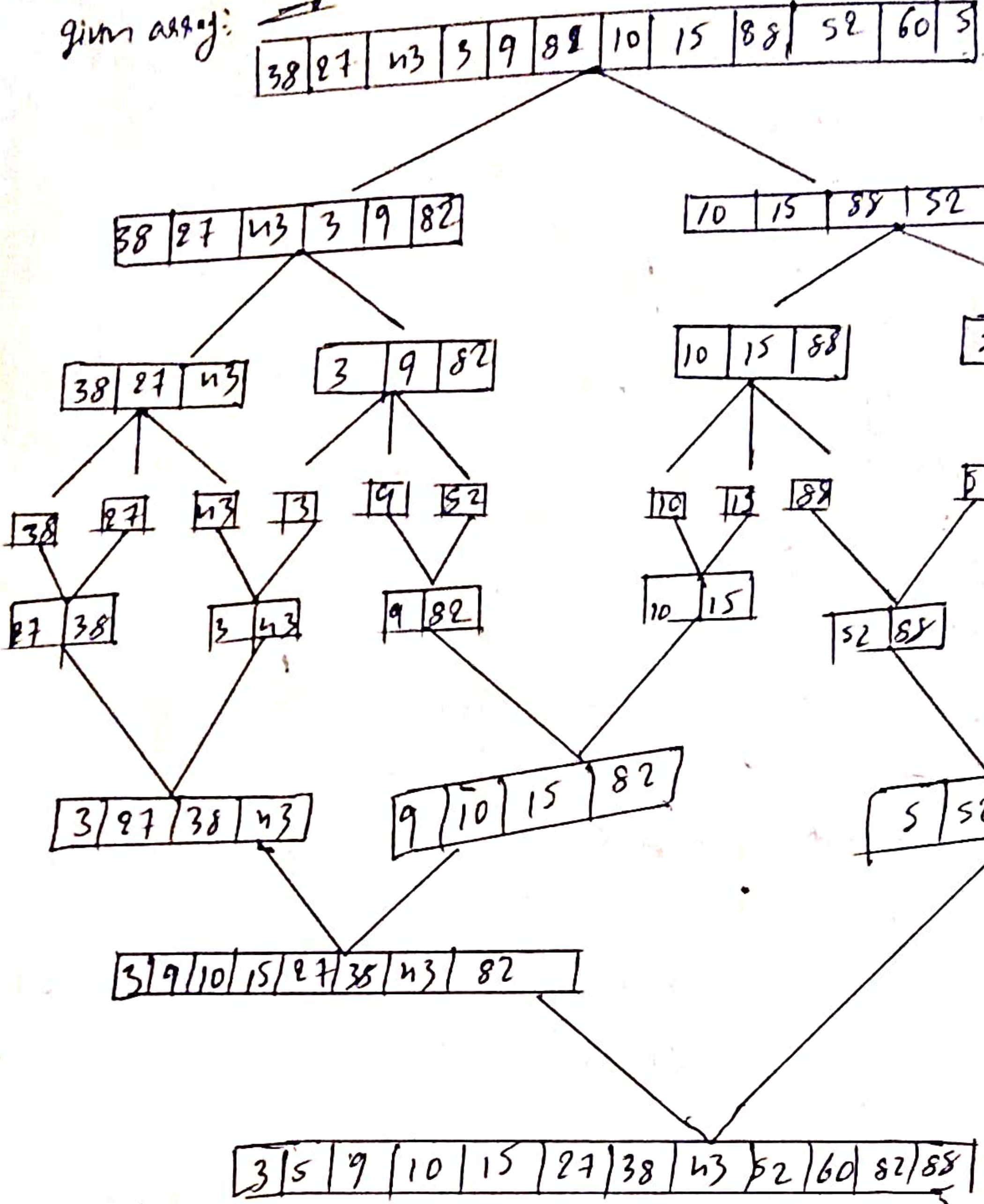
Assignment- 15

10) Sort the following elements using merge sort divide and Conquer stategy {38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5}; using and analyse time complexity of the algorithm?

Solution:

merge sort

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) where n is the num. of elements in the list. this is because the list is split into halves log n times and n merging sor..is ofintegriall the elements at each level takes o(n) time.

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

Solution given array = $\boxed{64|34|25|12|22|11|90}$

In bubble sort we bring the smallest element in there correct position continue this until each element reach there 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	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	64	25	34	90
11	12	22	25	64	34	90
11	12	22	25	34	64	90

∴ The sorted array is

(11, 12, 22, 25, 34, 64, 90)

Selection sort complexity

Selection sort is an another simple comparison sorted algorithm.

Best case: $O(n^2)$

Average case: $O(n^2)$

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

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?

Solution.

given: array: | 64 | 25 | 12 | 22 | 11 |

In the selection we will fix the lar. from the largest elements in those 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 Comparision sorted algorithm

Best case: $o(n^2)$

Avrag case: $o(n^2)$

worst case: $o(n^2)$

The selection sort has a time complexity $o(n^2)$ it always goes through the same n. of comparisons

4) 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 elements using insertion sort using Brute force algorithm strategy analyze time complexity.

Solution

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

To sorting array using insertion sort. insert one by one in the array and sort the array while inserting. from 1st to last elements

⇒Insert 4

4

insert -2, -2<4 so

-2	4

insert 5, 5>4 then

-2	4	5

insert 3, 3 is 3<4 & 3>-2

-2	3	4	5

insert 10, 10>5 then

-2	3	4	5	10

insert , -5 . -5<-2 then

-5	-2	3	4	5	10

insert 2, 2>-2 & 2<3 then

-5	-2	2	3	4	5	10

insert 8 . 8>5 & 8<10 then

-5	-2	2	3	4	5	8	10

insert -3, -3<-2 & -3>-5 then

-5	-3	-2	2	3	4	5	8	10

insert 6 . 6>5 & 6<8 then

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

insert 7 , 7>6 & 7<8 then

-5	-3	-2	2	3	4	5	6	7	8	10

insert -4 . -4>-5 & -4< -3 then

-5	-4	-3	-2	2	3	4	5	6	7	8	10

insert 1 , 1<2 & 1>-2 then

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

insert 9 , 9<10 & 9>8 then

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

insert -1, -1>-2 &, -1<1 then

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

insert 0 , 0<1 & 0>-1 then

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

insert -6 . -6 < -5 & ~~-6>~~4 then

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

insert -8 , -8< -6 then

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

insert 11 11>10 then

-8	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11

∴ The sorted array is

-8	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11

Time Complexity analysis

Best Case: O(n) - This occurs when the array is already sorted the inner loop runs zero times for every element

Average case: O(n²) - This happens because, on average, the algorithm will have to move half of the element for each insertion

Worst Case: O(n²) - This occurs when the array is sorted in reverse order each insertion takes O(n) time.

The insertion sort has a time Complexity of O(n²)

3) 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.

Solution

given array = [38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5]

insert 38

38

insert 27, 27 ≮ 38 then

27	38

insert 43 43 > 38 then

27	38	43

insert 3, 3 < 27

3	27	38	43

insert 9

3	9	27	38	43

Insert 82, 82 > 43 then

3	9	27	38	43	82

Insert 10 ,10>9 & 10<27 then

3	9	10	27	38	43	82

insert 15 ,15>10 & 15<27 then

3	9	10	15	27	38	43	82

insert 88 ,88>82 then

3	9	10	15	27	38	43	82	88

insert 52 . 52>43 & 52<88

3	9	10	15	27	38	43	52	82	88

insert 60 , 60 <82 & 60>52

3	9	10	15	27	38	43	52	60	82	88

insert 5 , 5>3 & 3<9 then

3	5	9	10	15	27	38	43	52	6	82	88

∴ The sorted array is

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

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: $O(n^2)$ This happens because, on average, the 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

The insertion sort has a time complexity of $O(n^2)$