P.V.S.Sucharitha
192311103
CSA0669.

Analytical Assignment:-02 ①

(20.) 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 follocing Elements using insertion sort using Brute force approach strategy analyze complexity of algorithm.

Given array.

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

⇒ 4 -2 5 3 10 -5 2 8 -3 6 7 -4 1 9 -1 0 -6 -8 11 -9.
i j

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

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

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

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

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

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

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

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

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

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

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

⇒ -5 -2 3 4 5 2 8 -3 6 7 10 -4 1 9 -1 0 -6 -8 11 -9.
i j

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

i j

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

i j

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

i i j

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

i j i

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

Sorted.

Time complexity:-

Best case (O(n)) - This occurs when the array is already sorted. The inner loop will run only once for each element.

Average case:- O(n²) - The list is randomly order.

Worst case:- O(n²): If the list is in reverse order

(19) Sort the following element using insertion sort using Brute Force approach Strategy [38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5] and analyze complexity algorithm

–m

Given that

Given that

1) [38 27 43 3 9 82 10 15 88 52 60 5]
i j

2) 27 38 43 3 9 82 10 15 88 52 60 5
i i j

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

4) 27 3 38 43 9 82 10 15 88 52 60 5
i j

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

6) 3 27 38 9 43 82 10 15 88 52 60 5
i i j

7) 3 27 9 38 43 82 10 15 88 52 60 5
i j

8) 3 9 27 38 43 82 10 15 88 52 60 5
i j

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

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

11) 3 9 27 10 38 43 82 15 88 52 60 5
i j

12) 3 9 10 27 38 43 82 15 88 52 60 5
i j

13) 3 9 10 27 38 43 15 82 88 52 60 5
i j

14) 2 9 10 27 38 15 43 82 88 52 60 5

15) 3 5 9 10 25 27 38 43 52 60 82 88

Time complexity:-

Best case - $O(n)$ - This occurs when the array is already sorted the inner loop will run only once.

Avg case - $O(n^2)$ - The list is randomly orderd

Worst case - $O(n^2)$ - If the list is in reverse

Space complexity:-

$O(1)$ - Insertion sort

(18) Sort the array 64, 25, 12, 22, 11 using selection Sort what is time complexity of selection Sort in the best, worst, average cases.

Given that,

0	1	2	3	4
64	25	12	22	11

↑ Sort ↑ min.

Time complexity
Best case : $O(n^2)$
Avg case : $O(n^2)$
worst case: $O(n^2)$

11	25	12	22	64

Sorted ↑ start ↑ min

11	12	25	22	64

Sorted ↑ start ↑ min

11	12	25	22	64

(17) Sort the array 64, 34, 25, 12, 22, 11, 90 using Bubble Sort What is time complexity of Bubble Sort in best, worst, average cases.

Given that,

64	34	25	12	22	11	90
i	j					
34	64	25	12	22	11	90
	i	j				
34	25	64	12	22	11	90
		i	j			
34	25	12	64	22	11	90
			i	j		
34	25	12	22	64	11	90
				i	j	
34	25	12	22	11	64	90
					i	j

It-2

34	25	12	22	11	64	90
i	j					
25	34	12	22	11	64	90
	i	j				
25	12	34	22	11	64	90
		i	j			
25	12	22	34	11	64	90
			i	j		

$25 \quad 12 \quad 22 \quad 11 \quad 34 \quad \underset{i}{64} \quad \underset{j}{90}$

It-3

$\underset{i}{25} \quad \underset{j}{12} \quad 22 \quad 11 \quad 34 \quad 64 \quad 90$

$12 \quad \underset{i}{25} \quad \underset{j}{22} \quad 11 \quad 34 \quad 64 \quad 90$

$12 \quad 22 \quad \underset{i}{25} \quad \underset{j}{11} \quad 34 \quad 64 \quad 90$

$12 \quad 22 \quad 11 \quad \underset{i}{25} \quad \underset{j}{34} \quad 64 \quad 90$

$12 \quad 22 \quad 11 \quad 25 \quad \underset{i}{34} \quad \underset{j}{64} \quad 90$

$12 \quad 22 \quad 11 \quad 25 \quad 34 \quad 64 \quad 90.$

16. Sort the following Elements using Merge sort divide and conquer strategy [38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5] and analyze complexity of algorithm.

It-4

$\underset{i}{12} \quad \underset{j}{22} \quad 11 \quad 25 \quad 34 \quad 64 \quad 90$

$12 \quad \underset{i}{22} \quad \underset{j}{11} \quad 25 \quad 34 \quad 64 \quad 90$

$12 \quad 11 \quad \underset{i}{22} \quad \underset{j}{25} \quad 34 \quad 64 \quad 90$

$12 \quad 11 \quad 22 \quad \underset{i}{25} \quad \underset{j}{34} \quad 64 \quad 90$

$12 \quad 11 \quad 22 \quad 25 \quad \underset{i}{34} \quad \underset{j}{64} \quad 90$

$12 \quad 11 \quad 22 \quad 25 \quad 34 \quad \underset{i}{64} \quad \underset{j}{90}$

It-5

$\underset{i}{12} \quad \underset{j}{11} \quad 22 \quad 25 \quad 34 \quad 64 \quad 90$

$11 \quad \underset{i}{12} \quad \underset{j}{22} \quad 25 \quad 34 \quad 64 \quad 90$

$11 \quad 12 \quad 22 \quad \underset{i}{25} \quad \underset{j}{34} \quad 64 \quad 90$

$11 \quad 12 \quad 22 \quad 25 \quad \underset{i}{34} \quad \underset{j}{64} \quad 90$

$11 \quad 12 \quad 22 \quad 25 \quad 34 \quad \underset{i}{64} \quad \underset{j}{90}$

$11 \quad 12 \quad 22 \quad 25 \quad 34 \quad \underset{i}{64} \quad \underset{j}{90}.$

Time complexity:

Best - $O(n)$

Avg - $O(n^2)$

Worst - $O(n^2)$

16) sort the following Elements using Merge sort divide & conquer strategy [38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5] and analyze complexity of algorithm.

Given that,

0	1	2	3	4	5	6	7	8	9	10	11
38	27	43	3	9	82	10	15	88	52	60	5

$M = \frac{l+h}{2} = \frac{0+11}{2} = 5.5 = 6$

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

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

$M = \frac{l+h}{2} = \frac{0+6}{2} = 3$ $\qquad M = \frac{l+h}{2} = \frac{7+11}{2} = 9$

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

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

$M = \frac{l+h}{2} = \frac{0+3}{2} = 2$ $\quad M = \frac{l+h}{2} = \frac{4+6}{2} = 5$ $\quad M = \frac{7+9}{2} = 8$ $\quad M = \frac{21}{2} = 10$

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

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

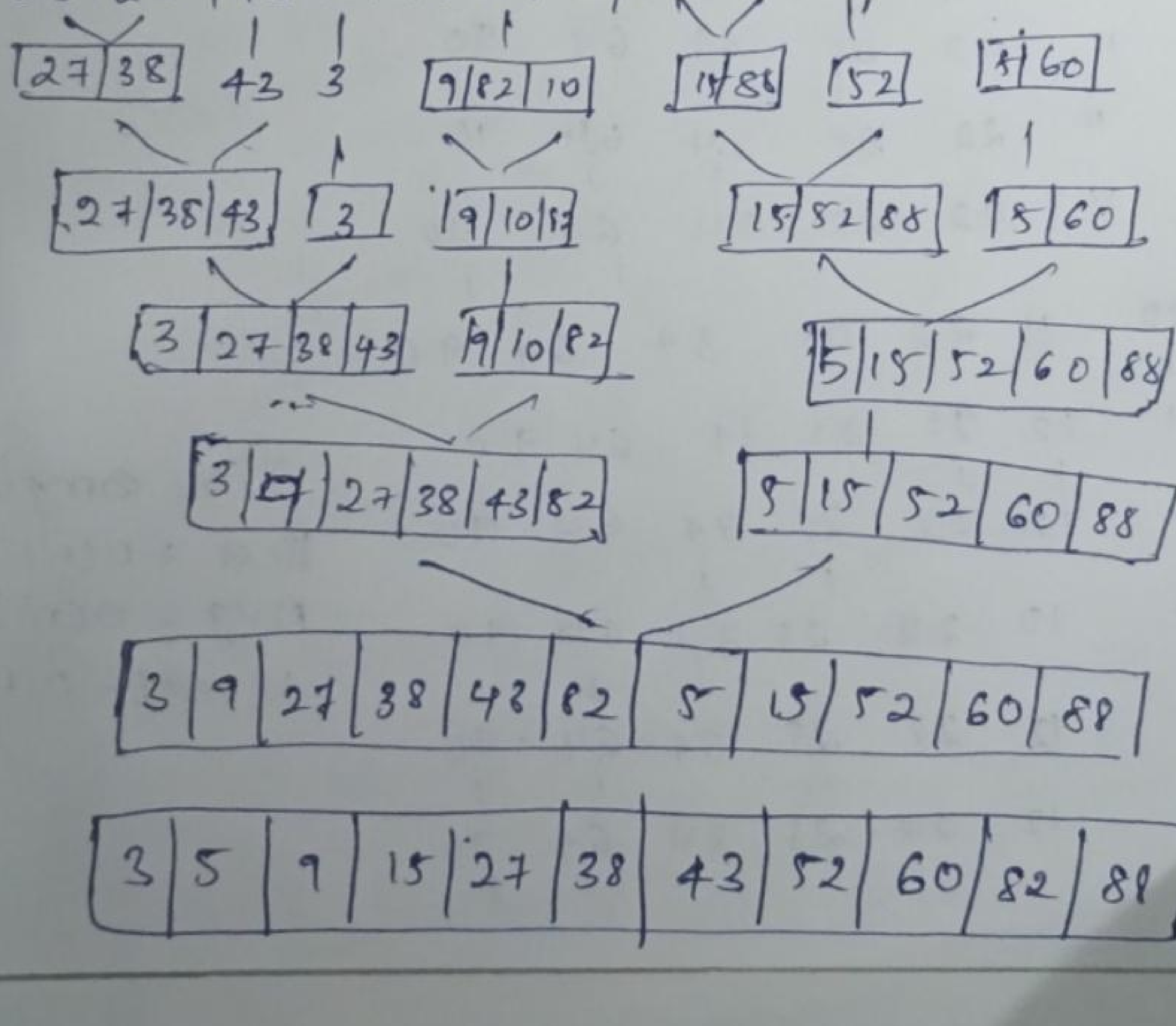
$M = \frac{0+2}{2} = 1$

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

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

M=0

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



sorted.

Time complexity:-

Best - $O(n^2)$

Avg - $O(n^2)$

worst - $O(n^2)$

(15) Find the index of the target value 10 using binary search from the following list of elements

[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

Given that

0	1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18	20

$M = \frac{l+h}{2} = \frac{0+9}{2} = 4.5 = 5$ (or) 5

0	1	2	3	$\boxed{4}$	5	6	7	8	9
2	4	6	8	$\boxed{10}$	12	14	16	18	20
				Mid					

Target = 10

a[mid] = Target

∴ 10 = 10

∴ Target found.

(14) Find the no. of times to perform swapping for selection sort Also Estimate the time complexity for the order of notation sets (12, 7, 5, -2, 18, 6, 13, 4)

S = 12, 7, 5, -2, 18, 6, 13, 4

1)	12	7	5	-2	18	6	13	4
	↓			↓				
	Start			min				

2.)	-2	7	5	12	18	6	13	14
		↓	↓					
		Start	min					

3.)	-2	5	7	12	18	6	13	14
			↓			↓		
			Start			min.		

4.)	-2	5	6	12	18	7	13	14
					↓	↓		
					st	min		

5.)	-2	5	6	7	18	12	13	14
					↓	↓		
					st	min		

$T(n) = 2T(\frac{n}{2}) + c(n)$

6) -2 5 6 7 12 18 13 14

$\overset{\downarrow}{\text{St}}$ (18) $\overset{\downarrow}{\text{min}}$ (13)

7) -2 5 6 7 12 13 18 14

$\overset{\downarrow}{\text{Start}}$ (18) $\overset{\downarrow}{\text{min}}$ (14)

8)

-2	5	6	7	12	13	14	18

Time complexity:

Best - $O(n^2)$

Avg - $O(n^2)$

Worst - $O(n^2)$

Space complexity = $O(1)$

Total No of Swaps = 6

(13) Apply Merge sort and order the list of 8 Ele Data d=(45, 67, -12, 5, 22, 30, 50, 20). Set up a recurrence relation for the no of key comparsions Made by Merge sort.

$d = \overset{0}{45}, \overset{1}{67}, \overset{2}{-12}, \overset{3}{5}, \overset{4}{22}, \overset{5}{30}, \overset{6}{50}, \overset{7}{20}$

$M = \frac{0+7}{2} = 4$

$\overset{0}{45}\ \overset{1}{67}\ \overset{2}{-12}\ \overset{3}{5}\ \overset{4}{22}\ |\ \overset{5}{30}\ \overset{6}{50}\ \overset{7}{20}$

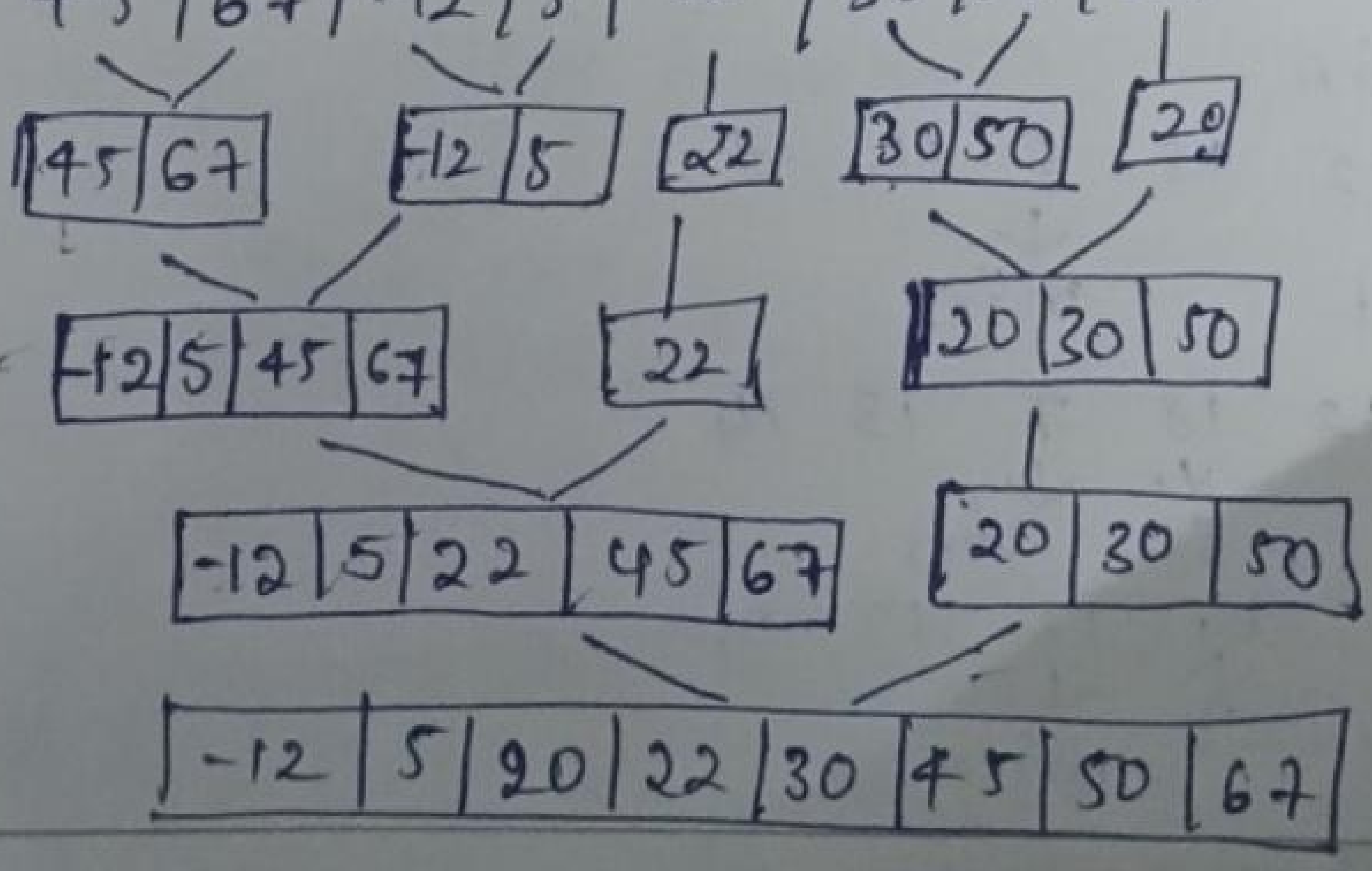
$M = \frac{0+4}{2} = 2$

$\overset{0}{45}\ \overset{1}{67}\ \overset{2}{-12}\ |\ \overset{3}{5}\ \overset{4}{22}\ |\ \overset{5}{30}\ \overset{6}{50}\ |\ \overset{7}{20}$

$M = \frac{0+2}{2} = 1$

$\overset{0}{45}\ \overset{1}{67}\ |\ \overset{2}{-12}\ |\ \overset{3}{5}\ |\ \overset{4}{22}\ |\ \overset{5}{30}\ |\ \overset{6}{50}\ |\ \overset{7}{20}$

45 | 67 | -12 | 5 | 22 | 30 | 50 | 20



Recurrence Relation:-

$T(n) = 2T\left(\frac{n}{2}\right) + c(n)$ $\quad\therefore \theta(n^k \log_n^{p+1})$

$a=2, \quad k=1$ $\quad \theta(n^1 \log_n^2)$

$b=2, \quad p=1$

$\log_b^a = \log_2^2 = 1$ $\quad \therefore \boxed{\theta(n \log n)}$

$\Rightarrow \boxed{\log_b^a = k}$

12.) Demonstrate Binary Search Method to Search key = 23, from the array arr[] = [2, 5, 8, 12, 16, 23, 38, 56, 72, 91].

arr[] = [2, 5, 8, 12, 16, 23, 38, 56, 72, 91]

key = 23.

0	1	2	3	4	5	6	7	8	9
2	5	8	12	16	23	38	56	72	91

$M = \frac{l+h}{2} = \frac{0+9}{2} = 4.5 = 5$

0	1	2	3	4	5	6	7	8	9
2	5	8	12	16	$\boxed{23}$	38	56	72	91
					Mid				

∴ arr[mid] = 23

arr[mid] = key

23 = 23

∴ key is found.

11) Give an array of [4, -2, 5, 3, 10, -5, 2, 8, -3, 6, 7, -4, 1, 9, -1, 0, -6, -8, 11, -9] integers, find Max & min product that can be obtained by Multiplying 2 integers from array.

Given that,

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

Maximum Product:-

2 largest no's : 11, 10

2 smallest (-ve no's) : -9, -8

Products:-

$11 \times 10 = 110$

$-9 \times -8 = +72$

∴ Max product = 110

Minimum Product:-

$\cdot 11 \times -9 = -99$

$10 \times -9 = -90$

$\therefore$ Min product $= -99$.

(10) Solve the following recurrence relations & find the order of growth for solutions.

$T(n) = 4T(n/2) + n^2 T(1) = 1$

Given that,

$T(n) = 4T(n/2) + n^2 T(1) = 1$

By Master's Theorm

$a = 4 \quad k = 2 \quad p = 0$

$b = 2 \quad f(n) = n^2$

$\log_b^a = \log_2^4 = \log_{2^1}^{2^2} = \log_1^2 = 2$

$\therefore \log_b^a = k$

$p > -1$, so,

$= \theta(n^k \log_n^{p+1})$

$= \theta(n^2 \log^1 n)$

$= \theta(n^2 \log n) = T(n)$

(9) Determaine Whether $h(n) = n \log n + n$ in $\theta(n \log n)$

Prove a rigorous proof your conclusion.

Given $h(n) = n\log n + n$

$C_1 \cdot n \log n \leq n \log n + n \leq C_2 \cdot n \log n$

upper bond:-

$n \log n + n \leq C_2 \cdot n \log n$.

$n \log n + n \leq n \log n + n \log n = 2n \log n$

$C_2 = 2$

$n \log n + n \leq 2n \log n$

lower bound:-

$C_1 \cdot n \log n \leq n \log n + n$

$C_1 \cdot n \log n \leq n \log n + n$

divide both sides by (n)

$C_1 \cdot \log n \leq \log n + 1$

$\frac{1}{2} \log n \leq \log n$.

$\therefore h(n) = n\log n + n \in \Theta(n\log n)$

(8) let $f(n) = n^3-2n^2+n$ & $g(n)=n^2$ show whether $f(n)=\Omega(g(n))$ is true or false and justify your answer

Given that,

$$f(n) = n^3-2n^2+n$$
$$g(n)=n^2$$
$$f(n) \geq c\cdot g(n)$$
$$f(n) = n^3-2n^2+n$$
$$= n^2(n-2)+n$$
$$= n^2(n-2+\frac{1}{n})$$

Compare $f(n)$ & $g(n)$:

$$f(n) = n^2(n-2+\frac{1}{n}) \geq c\cdot n^2$$
$$n^2(n-2+\frac{1}{n}) \geq c\cdot n^2$$
$$n^2(n-2+\frac{1}{n}) + c\cdot n^2 \geq 0$$
$$n^2(n-2+\frac{1}{n}+c) \geq 0$$
$$\therefore n-2+\frac{1}{n}+c \geq 0$$

This inequality is not always true for example, When n is close to $2\cdot n-2+\frac{1}{n}+c$ can be neg

$$\therefore f(n) \neq \Omega(g(n))$$

(7) Big Theta Notation: Determine whether $h(n)=4n^2+3n$ is $\Theta(n^2)$ or not.

Given that,

$$h(n) = 4n^2+3n$$

First, we need to find the constant c such that $h(n) \geq c\cdot n^2$ for large enough n.

$$h(n) = 4n^2+3n$$
$$= n^2(4+\frac{3}{n})$$
$$h(n) = n^2(4+\frac{3}{n}) \geq c\cdot n^2$$
$$\Rightarrow n^2(4+\frac{3}{2}) \geq c\cdot n^2$$
$$\Rightarrow 4+\frac{3}{n} \geq c$$

This inequality to hold for all n, we need $4+\frac{3}{n} \geq c$ for all n.

this inequality is not always true when n is close to $0.4+\frac{3}{4}$ can be less than c.

∴ We can't find a constant c such that

$$h(n) \geq c \cdot n^2$$

$$\therefore \boxed{h(n) \neq \Theta(n^2)}$$

(6) Big Omega Notation: Prove that $g(n) = n^3+2n^2+4n$ is $\Omega(n^3)$

Given that

$$g(n) = n^3+2n^2+4n$$

$$g(n) \geq c \cdot n^3$$

$$g(n) = n^3+2n^2+4n$$

$$= n^2(n+2)+4n$$

$$g(n) \geq n^3$$

$$g(n) = n^2(n+2)+4n \geq c \cdot n^3$$

$$n^2(n+2)+4n \geq c \cdot n^3$$

$$n^2(n+2)+4n-c \cdot n^3 \geq 0$$

$$n^2(n+2)+4n-c \cdot n^3 \geq 0$$

$$n^2(n+2)+4n-c \cdot n^3 \geq 0$$

∴ This inequality is not always true when n is close to $0 \cdot n^2(n+2)+4n-c \cdot n^3$ can be (-ve)

$$\therefore g(n) \neq \Omega(n^3)$$

(5) Big O Notation: ST $f(n) = n^2+3n+5$ is $O(n^2)$

$$f(n) \leq c \cdot n^2$$

$$f(n) = n^2+3n+5 = n^2+3n+5$$

$$f(n) = n^2+3n+5 \leq c \cdot n^2$$

$$n^2+3n+5 \leq c \cdot n^2$$

$$n^2+3n+5 \leq c \cdot n^2$$

$$3n+5 \leq c \cdot n^2$$

$$f(n) = n^2+3n+5 \in O(n^2).$$

$\therefore f(n) - n^2+3n+5 \in O(n^2)$

(4) $T(n) = \begin{cases} 2T(n-1) & \text{if } n>0 \\ 1 & \text{otherwise} \end{cases}$

$T(n) = 2T(n-1)$

$T(n-1) = 2[2T(n-2)] = 2^2 T(n-2)$

$T(n) = 2^2[2T(n-3)] = 2^3 T(n-3)$

$T(n) = 2^k T(n-k)$

$n-k=0, n=k$

$T(0)=1$

$T(n) = O(2^n)$ //.

(3) $T(n) = \begin{cases} 2T(\frac{n}{2})+1 & \text{if } n>1 \\ 1 & \text{otherwise} \end{cases}$

$T(n) = aT(n/b) + f(n)$

if $f(n) = O(n^{\log_b^a - E})$

Then $T(n) = \theta(n^{\log_b^a})$

if $f(n) = \theta(n^{\log_b^a} \log_n^k)$

then $T(n) = \theta(n^{\log_b^a} \log_n^{k+1})$

if $f(x) = \Omega(n^{\log_b^a + E})$

then $T(n) = \theta(f(x))$

$T(n) = 2T(n/2)+1$

$a=2, b=2.$

$k=1, p=1$

$\log_a^b = \log_2^2 = 1$

$\log_a^b = k$

$p \geq -1 \Rightarrow \theta(n^k \log_n^{p+1})$

$\theta(n^1 \log_n^2)$

$\theta(n \log n)$

(1) If $t_1(n) \in O(g(n))$ and $t_2(n) \in (g^2(n))$, then $t_1(n) + t_2(n)$ $\in (O max(g'(n)), g_2(n))$, prove that assertions.

Givan that,

$t_1(n) \leq C_1 g_1(n)$

$t_2(n) \leq C_2 g_2(n)$

Consider $t_1(n) + t_2(n)$

$t_1(n) + t_2(n) \leq C_1 g_1(n) + C_2 g_n(n)$

Find an upper bound for $t_1(n) + t_2(n)$

$\max\{g_1(n), g_2(n)\} \geq g_1(n)$ and

$\max\{g_2(n), g_2(n)\} \geq g_2(n)$

$\therefore t_1(n) + t_2(n) \leq C_1$

$\max\{g_1(n), g_1(n)\} + C_2$

$\max\{g_1(n); g_2(n)\}$

let $C = C_1 + C_2$ then

$t_1(n) + t_2(n) \leq C$

$\max\{g_1(n), g_2(n)\}$

let $C = C_1 + C_2$ then.

$t_1(n) + t_2(n) \leq C$

$\max\{g_1(n), g_2(n)\}$

Thus,

$t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$

$|t_1(n) + t_2(n)| \leq C$

$\max\{g_1(n), g_2(n)\}$

Thus, The Statement is proven.