Y.Harika
192325023
CSA0669

1

ANALYTICAL QUESTION-2

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 following 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 swap

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

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

-2 4 3 5 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 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

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

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

i j

-9 -8 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7 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: $O(n^2)$ - The list is randomly ordered

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

19. 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 algorithm.

Given

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

i j

1) 27 38 43 3 9 82 10 15 88 52 60 5

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

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

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

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

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

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

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

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

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

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

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

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

Sorted.

Time Complexity:

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

Avg Case - O(n²)) - The list is randomly ordered.

Worst Case - O(n²)) - 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

0	1	2	3	4
64	25	12	22	11

↑ start (64) ↑ min (11)

11	25	12	22	64

Sorted (11) ↑ Start (25) ↑ Min (12)

11	12	25	22	64

Sorted (11, 12) ↑ Start (25) ↑ Min (22)

11	12	22	25	64

Sorted

Time complexity

Best Case: $O(n^2)$

Avg case: $O(n^2)$

Worst Case: $O(n^2)$

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

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 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	12	22	11	34	64	90
					i	j

It-3

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

V.Harika

8

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.

Given

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

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

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

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

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

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

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

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

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

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

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

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

It 4

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

It. 5

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

Sorted.

Time complexity:

Best – $O(n)$

Avg – $O(n^2)$

Worst – $O(n^2)$

Y.Harika

3	9	27	38	43	82

5	15	52	60	88

3	9	27	38	43	82	5	15	52	60	88

3	5	9	15	27	38	43	52	60	82	88

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

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

$M = \frac{l+h}{2} = \frac{0+9}{2} = 4.5 \approx 5$ (or) 4

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

Target = 10

a[mid] = Target

$\therefore$ 10 = 10

$\therefore$ Target found,,

14) Find the no of times to perform swap selection sort. Also estimate the time complexity the order of notation set S (12, 7, 5, -2, 18, 6, …

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

1) 12 7 5 -2 18 6 13 4

Start (12), min (-2)

2) -2 7 5 12 18 6 13 14

Start (7), min (5)

3) -2 5 7 12 18 6 13 14

Star (7), min (6)

4) -2 5 6 12 18 7 13 14

Sta (12), Min (7)

5) -2 5 6 7 18 12 13 14

St (18), min (12)

6) -2 5 6 7 12 18 13 14

start (18), min (13)

7) -2 5 6 7 12 13 18 14

Strt (18), Min (14)

8)

-2	5	6	7	12	13	14	18

$\underbrace{\hspace{6em}}_{\text{sorted}}$

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 number of key comparsions Made by mergesort.

	0	1	2	3	4	5	6	7
d =	45,	67,	-12,	5,	22,	30,	50,	20

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

0	1	2	3	4	|	5	6	7
45	67	-12	5	22	|	30	50	20

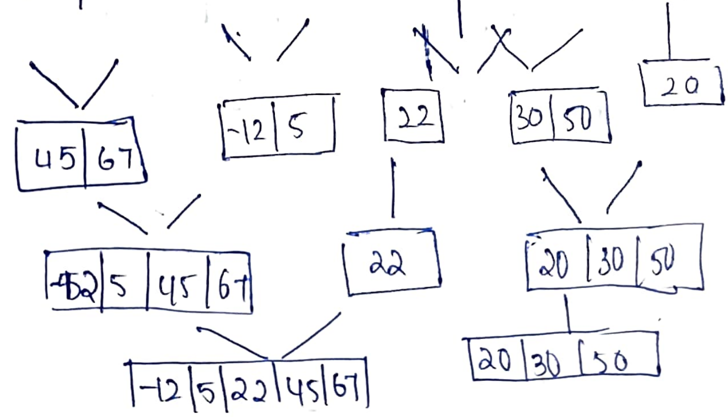
$M = \frac{0+4}{2} = 2$ $\qquad$ $M = 6$

0	1	2	|	3	4	|	5	6	|	7
45	67	-12	|	5	22	|	30	50	|	20

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

0	1	|	2	|	3	|	4	|	5	|	6	|	7
45	67	|	-12	|	5	|	22	|	30	|	50	|	20

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



-12	5	22	45	67

20	30	50

-12	5	20	22	30	45	50	67

Sorted.

Recurrence Relation:

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

$a = 2$, $k = 1$

$b = 2$, $p = 1$

$$\log_b a = \log_2 2 = 1$$

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

$$\therefore \Theta(n^k \log^{p+1} n)$$

$$\Theta(n^1 \log^2 n)$$

$$\boxed{\therefore \Theta(n \log n)}$$

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

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

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) 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, find Maximum and Minimum product that can be obtained by Multiplying 2 integers from array.

Given

[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:

$11 \times -9 = -99$

$10 \times -9 = -90$

$\therefore$ Min product $= -99$

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

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

Given

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

By Master's Theorem.

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

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

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

$\therefore \log_b a = k$

$p > -1$, so,

$= \Theta(n^k \log^{p+1} n)$

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

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

9. Determine 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:

$n \log n + n \leq c_2 \cdot n \log n$

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

$c_2 = 2$

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

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)

$c_1 \cdot \log n \leq \log n + 1$

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

$\therefore \boxed{c_1 \cdot n\log n \leq n\log n + n \leq c_2 \cdot n\log n}$

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

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

Given $f(n) = n^3 - 2n^2 + n$

$g(n) = n^2$

$f(n) \geq c \cdot g(n)$

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

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

$= n^2(n - 2 + \frac{1}{n})$

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. $n-2+\frac{1}{n}+c$ 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

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

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

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

$= n^2(4+\frac{3}{n})$

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

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

$\Rightarrow 4+\frac{3}{n} \geq c$

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

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

$\therefore$ 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

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

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

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

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

$$g(n) \epsilon\, n^3$$

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

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

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

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

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

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

$$\boxed{\therefore g(n) \neq \Omega(n^3).}$$

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

Given

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

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

for all $n > n_0$

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

$= n^2 + 3n + 5$

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

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

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

$3n+5 \le c\, n^2$

$\therefore$ when n is close to 0, $3n+5 \le c \cdot n^2$ can be (-ve)

$f(n) = O(n^2)$

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

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

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

$= 2^2 T(n-2]$

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

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

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

$n-k = 0, \; n=k$

$T(0) = 1$

$$\boxed{T(n) = O(2^n)}$$

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 $f(n) = O(n \log_b^a - \epsilon)$

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 $f(n) = \Omega(n \log_b^a + \epsilon)$

then $T(n) = \Theta(f(n))$

2) ~~Find~~ ~~time~~ ~~complexity~~ ~~Complexitu~~

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

$a = 2$

$b = 2$ $\quad k=1 \;, P=1$

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

$\log_a^b = k$

$P \geq -1 \qquad \Theta(n^k \log_n^{P+1})$

$O(n' \log^2 n)$

$\Rightarrow O(n \log n)$

1) If $t_1(n) \in O(g_1(n))$ and $t_2(n) \in O(g_2(n))$ then Prove $t_1(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$.

the assertions,:

$f_1(n) \leq c_1 g_1(n)$ for all $n \geq n_0$

$f_2(n) \leq c_2 g_2(n)$ for all $n \geq n_0$

Adding

$f_1(n) + f_2(n) \leq c_1 g_1(n) + c_2 \cdot g_2(n)$

Since

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

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

$f_1(n) + f_2(n) \leq c_1 \max\{g_1(n), g_2(n)\} + c_2 \max\{g_1(n), g_2(n)\}$

$\leq (c_1 + c_2) \max\{g_1(n), g_2(n)\}$

let $c = c_1 + c_2$

$f_1(n) + f_2(n) \leq c \max\{g_1(n), g(2)\}$ for all $n \geq n_0$

$\therefore f_1(n) + f_2(n) = O(\max\{g_1(n), g_2(n)\})$