CSA0669 - Design and Analysis of Algorithm for Polynomial Problem

NAME: Rida S
REG NO: 192321010
DATE: 08/06/2024

1) If $t_1(n) \in O(g_1(n))$ and $t_2(n) \in O(g_2(n))$, prove that $t(n) + t_2(n) \in O(\max\{g_1(n), g_2(n)\})$

For any four arbitrary real numbers a_1, b_1, a_2, b_2.

$$a_1 \le b_1 \text{ and } a_2 \le b_2$$

we have $a_1 + a_2 \le 2 \max(b_1, b_2)$

Since $t_1(n) \in O(g_1(n))$, then there exists some constant c_1 and non-negative integer n_1 such that

$$t_1(n) \le c_1 g_1(n) \text{ for all } n \ge n_1$$

Since $t_2(n) < O(g_2(n))$, then there exists same constant c_2 and nonnegative integer n_2 such that

$$\begin{aligned} t_2(n) + t_2(n) &\le c_1 g_1(n) + c_2 g_2(n) \\ &\le c_3 g_1(n) + c_3 g_2(n) \\ &\le c_3 \{g_1(n) + g_2(n)\} \\ &\le 2 c_3 \max\{g_1(n), g_2(n)\} \end{aligned}$$

Hence, $t_1(n) + t_2(n) \le O(\max\{g_1(n), g_2(n)\}$, with constants c and required by the O definition being $2c_3 = c$ $\max\{c_1, c_2\}$ and $\max\{n_1, n_2\}$ respectively.

Find the time complexity of the below recurrance Equation:

$$T(n) = \{2T(n/2) + 1 \text{ if } n > 1$$

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

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

$k = 0$
$\log_b^a > k$

case i,
$\Theta(n^{\log a})$
$\Theta(n^1)$
$\Theta(n)$

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

Backward Substitution:

$T(n) = 2T(n-1) \rightarrow (1)$ Initial $T(0) = 0$

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

Sub (2) in (1):
$T(n) = 2[2T(n-2)]$
$T(n) = 2^2 T(n-2) \rightarrow (3)$

Sub (4) in (3): $T(n) = 2^2[2T(n-3)]$
$T(n) = 2^3 T(n-3) \rightarrow (5)$

$n = n-3$ $T(n-3) = 2T((n-3)-1)$
$T(n-3) = 2T(n-4) \rightarrow (6)$

Sub (6) in (5): $T(n) = 2^3[2T(n-4)]$
$= 2^4 T(n-4) \rightarrow (7)$
$T(n) = 2^k T(n-k)$
$n-k = 0 \Rightarrow n = k$

$T(n) = 2^k \cdot 1$

$T(n) = 2^k$

$\therefore n = k$

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

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

To prove that $f(n) = n^2 + 3n + 5$ is $O(n^2)$ we need to find constants c and n_0 such that :

$f(n) \le c \cdot n^2$ for all $n \ge n_0$

for $n \ge 1$, $n^2 \ge n$ ... so on

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

$f(n) = n^2 + 3n + 5 \le 9n^2$ for $n \ge 1$

So, for $c = 9$ and $n_0 = 1$

$f(n) \le c \cdot n^2$ for all $n \ge n_0$

that proves $f(n)$ is $O(n^2)$

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

To prove that $g(n) = n^3 + 2n^2 + 4n$ is $\Omega(n^3)$ we need to find constants c and n_0 such that

$g(n) \ge c \cdot n^3$ for al $n \ge n_0$

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

for $n \ge 1$,

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

Since $2n^2$ and $4n$ are both less then n^3 when $n \ge$

So, for $c = 1$ and $n_0 = 1$

$g(n) \ge c \cdot n^3$ for all $n \ge n_0$

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

1) $h(n) = 4n^2 + 3n$ is $O(n^2)$.

For $n \geq 1$, $h(n) \leq 4n^2 + 3n^2$

(since $3n$ is less than the n^2 when $n \geq 1$

for this simplifies to $h(n) \leq 7n^2$

for $n \geq 1$

Therefore, $h(n)$ is $O(n^2)$

2) $h(n) = 4n^2 + 3n$ is $\Omega(n^2)$:

for $n \geq 1$, $h(n) \geq 4n^2$

(since $3n$ is +ive)

Therefore $h(n)$ is $\Omega(n^2)$

Since $h(n)$ is both $O(n^2)$ and $\Omega(n^2)$ it is $\Theta(n^2)$

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:

$n=1$: $f(1) = 1^3 - 2(1)^2 + 1$
$= 1 - 2 + 1$
$= 0$

$g(1) = -(-n)^2$
$= (-1)^2$
$= 1$

$n=2$: $f(2) = 3^3 - 2(3)^2 + 3$
$= 27 - 9 + 3$

$g(3) = (-3)^2$
$= 1$

$n=3$: $f(3) = 3^3 - 2(3)^2 + 3$
$= 27 - 9 + 3$
$= 21$

$n=4$: $f(4) = 4^3 - 2(4)^2 + 4$ $\quad g(n) = (-4)^2$
$= 64 - 32 + 4$ $\quad = 16$.
$= 32 + 4$
$= 36$

$n = 5$: $f(n) = 5^3 - 2(5)^2 + 5$ $\quad g(n) = (-5)^2$
$= 125 - 50 + 5$ $\quad = 25$
$= 75 + 5$
$= 80$

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

So it is best case according to asymptotic notation.

$f(n) = \Omega(g(n))$

) Determine whether $h(n) = n\log(n+n)$ is $\Theta(n\log n)$ prove a rigorous proof for your conclusion.

) Upper Bound: (O notation):

$h(n) \leq c_1 \cdot n\log n$ for all $n \geq n_0$

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

$\leq n\log n + n\log n$ (since $\log n$ is increasing)

$= 2n\log n$.

now, let $c_1 = 2$, then $h(n) \leq 2n\log n$ for all $n \geq$

So, $h(n)$ is $O(n\log n)$

) lower Bound: (Ω notation):

We need to find c_2 and n_0 such that

$h(n) \geq c_2 n\log n$ for all $n \geq n_0$

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

$\geq 1/2\, n\log n$ (for $n \geq 2$)

Now, let $c_2 = 1/2$, then $h(n) \geq 1/2 \cdot n\log n$
for all $n \geq 2$. So, $h(n)$ is $\Omega(n\log n)$

3) Combining Rounds:

Since $h(n)$ is both $O(n\log n)$ and $\Omega(n\log n)$
it is also $\Theta(n\log n)$

Thus, $h(n) = n\log n + n$ is in $\Theta(n\log n)$.

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

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

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

$a = 4$
$b = 2$ $\quad \log_b a = \log_2 4 = 2$

$K = 2$
$d = 2$
$\log_b a = K$

Case ii) $\quad p > -1 \; \Theta(n^K \log_n^{p+1})$

$\Theta(n^2 \cdot \log_n^{1+1})$

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

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

The order of growth for the solution is

$n^2 \cdot \log(n)$

Given an array of {9, -2, 5, 8, 10, -3, 2, 8, -3, 6, 7, -1, 1, 9, -1, 0, -6, -8, -11, -9} integers find maximum and minimum product that can be obtained :

Maximum Product :

2 largest no's : 11, 10
2 smallest (-ve no's) : -9, -8

Product : $11 \times 10 = 110$
$-9 \times -8 = +72$ ∴ max product = 110

Minimum Product : $11 \times -9 = -99$
$10 \times -9 = -90$
∴ Min product = -99

Answer : arr[] = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91}
Key = 23

2 5 8 12 16 23 38 56 72 91

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

2 5 8 12 16 | 23 | 38 56 72 91
mid

∴ arr[mid] = 23
arr[mid] = key
23 = 23
Key is found.

Answer :

d = 45, 67, -12, 5, 22, 30, 50, 20.

$M = 0+7/2 = 4$

45 67 -12 5 22 | 30 50 20

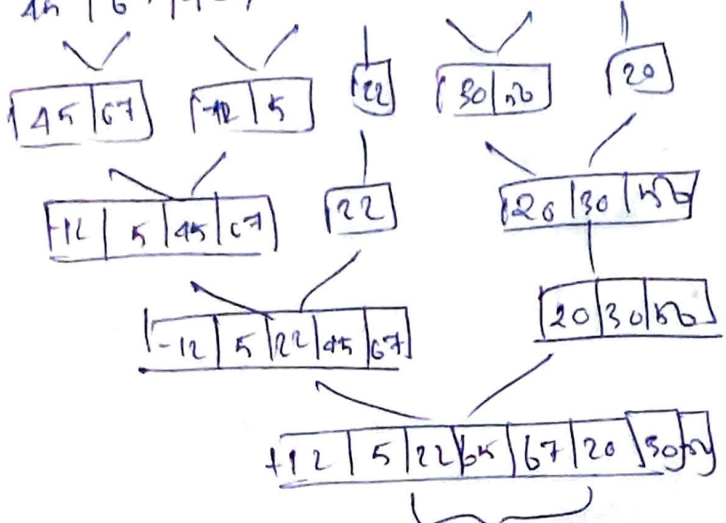
$M = 0+4/2 = 2$

45 67 | -12 | 5 22 | 30 50 70

$M = 0+2/2 = 1$

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

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



Recurrence Relation:

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

$a = 2, \; k = 1$

$b = 2, \; P = 1$

$\log_b a = k$

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

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

$\therefore \Theta(n \log n)$

Answer.

1) 12, 7, 5, -2, 18, 6, 13, 4

start min

2) -2, 7, 5, 12, 18, 6, 13, 4

start min

4) -2 5 6 12 8 7 13 14
start min

5) -2 5 6 7 18 12 13 14
start min

6) -2 5 6 7 12 18 13 14
start min

7) -2 5 6 7 12 13 18 14
start min

8) | -2 | 5 | 6 | 7 | 12 | 13 | 14 | 18 |

⇒ Sorted.

Time Complexity.
Best - $\Omega(n^2)$
Avg = $O(n^2)$
Worst $\Omega(n^2)$

Total no. of swaps = 6 //

Answer:

Given:

2 4 6 8 10 12 14 16 18 20

$m = l + n/2 = 0 + 9/2 = 4.5 \approx 5$ (or) 4.

2 4 6 8 10 12 | 14 16 18 20

Target = 10

a(mid) = Target

∴ 10 ≠ 10

⇒ Target found.

17) Answer: Given Array

I-1

64	34	25	12	22	11	90

34	64	25	12	22	11	90

34	25	12	64	22	11	90

34	25	12	22	64	11	90

34	25	12	22	11	64	90

I-2

34	25	12	22	11	64	90

25	34	12	22	11	64	90

25	12	22	34	11	64	90

25	12	22	11	34	64	90

I-3

25	12	22	11	34	64	90

12	25	22	11	34	64	90

12	22	25	11	34	64	90

12	22	11	25	34	64	90

I-4

12	22	11	25	34	64	90

12	22	11	25	34	64	90

12	11	22	25	34	64	90

15

12	11	22	25	34	64	90

11	12	22	25	34	64	90

Time complexity of selection sort:

Best - $O(n)$

AV - $O(n^2)$

worst = $O(n^2)$ //

) Answer:

given array

sorted array

64	25	12	22	11

unsorted array

i j

11	25	12	22	64

i j

11	12	25	22	64

i j

sorted array

11	12	22	25	64

Time Complexity.

Best : $O(n)$

Av : $O(n^2)$

worst : $O(n^2)$ //

Answer.

Answer:

= [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

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

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

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

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

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

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

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

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

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

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

Avg case $(O(n^2))$ The list is randomly ordered.

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