R. Harish Bharadwaj
1923210037

Assignment

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

i] $t_1(n) \in O(g_1(n)) \rightarrow$ there is a constant $c_1 > 0$ & $n \geq n_1$

$\therefore t_1(n) \leq c_1 \cdot g_1(n)$

ii] $t_2(n) \in O(g_2(n))$ $c_2 > 0$ & $n \geq n_2$

$t_2(n) \leq c_2 g_2(n)$

S.T. $t_1(n) + t_2(n) \in O(\max\{g(n), g_2(n)\}$

Proof:

~~$t_1(n) + t_2(n)$ & the~~

Let $g(n) = \max\{g_1(n), g_2(n)\}$

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

$t_1(n) \leq c_1 g_1(n) \leq c_1 \cdot g(n)$

$t_2(n) \leq c_2 \cdot g_2(n) \leq c_2 \cdot g(n)$

$t_1(n) + t_2(n) \leq c_1 \cdot g(n) + c_2 \cdot g(n) = (c_1 + c_2) \cdot g(n)$

Let $c = c_1 + c_2$

$\therefore t_1(n) + t_2(n) \leq c g(n)$

Since $g(n) = \max\{g_1(n), g_2(n)\}$

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

H.P.

2] Find $T(n)$ Time complexity

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

$2T(n/2)+1$ if $n>1$

$T(n) = aT(n/b) + f(n)$ $a=2$ $b=2$ $f(n)=1$

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

$f(n) = 1 \Rightarrow n^1 = n^k \Rightarrow k=1$

$\log_a^b = k$

Case 1:

$Tn = \Theta(n^k \log_a^b) = \Theta(n)$

ii) $T(n) = 2T(n-1)$ if $n>0$

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

$T(n-1) = 2T(n-2) \Rightarrow T(n) = 2^2 T(n-2)$

$T(n-2) = 2T(n-3) \Rightarrow T(n) = 2^3 T(n-3)$

$T(n) = 2T(n-1) + 2T(n-2) + 2T(n-3)$

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

Let $k=n$

$T(n) = 2^k T(0)$

$T(0) = 1 \Rightarrow T(n) = O(2^n)$

3] S.T: $f(n) = n^2+3n+5$ is $O(n^2)$

$O \rightarrow f(n) < c.g(n)$

$f(n) = n^2+3n+5$ $g(n) = n^4$

$n=1$ $f(n) = 9 > g(n) = 1$

$n=2$ $f(n) = 21 > 16$

$n=3$ $f(n) = 23 < 71$

$n=4$ $f(n) < g(n) \ldots$

$\therefore f(n) \leq g(n)$

$\therefore$ Big O is satisfied

6) $g(n) = n^3 + 2n^2 + 4n$ is $\Omega(n^3)$

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

$$n^3 + 2n^2 + 4n \geq c n^3$$

$n=1$ $g(n) = 7$ $c \cdot (n) = 1$

$n=2$ $g(n) = 24$ $c \cdot (n) = 8$

$n=3$ $g(n) = 57$ $c \cdot (n) = 27$

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

(H.P)

7] Determine $h(n) = 4n^2 + 5n$ is $O(n^2)$ or not

for $O(n^2)$

$$h(n) \leq c(n^2)$$

$$4n^2 + 5n \leq c(n^2)$$

$n=1$ $h(n) = 9$ $c(n^2) = (1)$

$n=2$ $h(n) = 26$ $c(n^2) = 4$

$$h(n) > c(n^2)$$

this shows $\Omega(n^2)$

$\therefore$ $h(n)$ is both $O(n^2)$ & $\Omega(n^2)$

8] $f(n) = n^3 - 2n^2 + n$ $g(n) = -n^2$ S.T $f(n) = \Omega(g(n))$

T.P if $f(n) \geq c \cdot g(n)$

$n=1$ $f(n) = 0$ $g(n) = 1$

$n=2$ $f(n) = 2$ $g(n) = -4$

$n=3$ $f(n) = 12$ $g(n) = -9$

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

$\therefore$ $\Omega(n)$ is proved

9] Determine $h(n) = n\log n + n$ is $O(n\log n)$.

~~To~~ $h(n) = n\log n + n$ to be $O(n\log n)$

then $h(n) \leq c \cdot (n\log n)$

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

$$n\log n + n = n(\log n + 1)$$

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

% by n

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

% by log n

$$\frac{\log n + 1}{\log n} \leq c_1$$

$$1 + \frac{1}{\log n} = c_1 = 2$$

To check $n\log n + n \leq 2(n\log n)$

$n=1$	$h(n) = 1$	$c(n\log n) = 0$
$n=2$	$h(n) = 2\log 2 + 2$	$c(n\log n) = 4\log 2$
$n=3$	$h(n) = 3\log 3 + 3$	$c(n\log n) = 6\log 3$

$\therefore h(n) < c \cdot g(n\log n)$

$\therefore$ we conclude $h(n)$ is $O(n\log n)$

10] Find order of growth

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

$a = 4 \quad b = 2$

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

$$f n = n^k \log_n^P = n^2$$

$$k = 2_n$$

$$\log_a^b = \log_4^2 = 2$$

$$\log_a^b = k$$

Case 2 :-

$P = 1$

$P > -1$

$$\therefore T(n) = n^k \log_n^{P-1}$$

$$= n^2 \log_n^1$$

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

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

Find max & min product that can be obtain by xing 2 digit

Sort

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

For max :-

i] $10 \times 11 = 110$

ii] $-9 \times -8 = 72$

Max :- (10, 11)

For Min :-

$-9 \times 11 = -99$

$-8 \times 11 = -88$

min = (-9, 11)

12] Demonstrate Binary Search method : search = 23
arr[ ] = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91}

Pseudocode:

```
def bs (arr, key);
    s = 0
    e = len(arr) - 1
    while (s <= e):
        mid = (s + e)//2
        if arr(mid) == key :
            return mid
        elif arr(mid) > key;
            s = mid+1
        else
            e = mid - 1
```

i] s = 0, e = 9, mid = 4
a[4] = 16 < 23 ∴ ~~e = mid~~ - 1 s = mid+1

ii] s = 5, e = 9, mid = 7
a[7] = 56 > 23 ∴ e = mid - 1

iii] s = 5, e = 6
mid = 5
a[5] = 23 = 23 Hence found.

13] d = (45, 67, -12, 5, 22, 30, 50, 80) do Merge Sort

[45, 67, -12, 5, 22, 30, 50, 80]

[45, 67, -12, 5] [22, 30, 50, 20]

[45,67,-12,5]

[45,67]   [-12,5]

⇓ Sorted   ⇓ Sorted

[-12,5,45,67]

[27,30,50,20]

[27,30]   [50,20]

⇓ Sorted   ⇓ [20,50]

[20,27,30,50]

[-12,5,20,27,30,45,50,67]

14] Find no. of time of swap in Selection Sort S = [12,7,5,-2,18,6,13,4]

i] [-2,7,5,12,18,6,13,4]

ii] [-2,4,5,12,18,6,13,7]

iii] [-2,4,5,6,18,12,13,7]

iv] [-2,4,5,6,7,12,13,18]

no. of swap = 4

Best case = $O(n^2)$ = Worst case

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

15] Key = 10   [2,4,6,8,10,12,14,16,18,20] binary Search

i] $s = 0$, $e = 9$, $m = (s+e)/2 = 4$

$a[4] = 10 = 10$

Hench found at 4th index

16] Sort using MergeSort [38, 27, 43, 3, 9, 82, 10, 15, 88, 52, 60, 5]

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

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

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

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

[27, 38] [43] sorted sorted sorted [5]

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

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

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

$T(n) = 2T(n/2)\,n$

$T(n) = O(n \log n)$

17] Sort [64, 34, 25, 12, 22, 11, 90] bubble Sort

1st Pass:- [34, 64, 25, 12, 22, 11, 90] [34, 25, 12, 22, 11, 64, 90]

[34, 25, 64, 12, 22, 11, 90]

[34, 25, 12, 64, 22, 11, 90]

[34, 25, 12, 22, 64, 11, 90]

2nd Pass : [25, 34, 18, 22, 11, 64, 90]

[25, 12, 34, 22, 11, 64, 90]

[25, 12, 22, 34, 11, 64, 90]

[25, 12, 22, 11, 32, 64, 90]

Best case: O(n²)

Worst case: O(n²)

3rd Pass = [12, 25, 22, 11, 32, 64, 90]

= [12, 22, 25, 11, 32, 64, 90]

[12, 22, 11, 25, 32, 69, 90]

4th Pass = [12, 11, 22, 25, 32, 69, 90]

5th Pass = [11, 12, 22, 25, 32, 69, 90]

18] Selection Sort [64, 25, 12, 22, 11]

1st Pass : [11, 25, 12, 22, 64]

2nd Pass : [11, 12, 25, 22, 64]

3rd Pass : [11, 12, 22, 25, 64]

4th Pass = [11, 12, 22, 25, 64] => Sorted

Best case : O(n²)

Worst Case : O(n²)

19] insertion Sort : [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, 9, 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, 52, 82, 88, 60, 5]

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

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

Best Case : O(n)

Worst case : O(n²)

20] [4, -2, 5, 3, 10, -5, 2, 8, -3, 6, 7, -4, 1, 9, -1, 0, -6, -8, 11, 9]
insertion sort.

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

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

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

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

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

Best case : $O(n)$

Worst cas : $O(n^2)$