ASSIGNMENT - 11

Challa. Ravi teja.
192325065 (CSA0670)

1. If $t_1(n) \in o(g_1(n))$ and $t_2(n) \in o(g_2(n))$; Then $t_1(n)+t_2(n) \in o(\max\{g_1(n), g_2(n)\})$ prove that assertions.

A: We need to show that $t_1(n)$ to $\max\{g_1(n), g_2(n)\}$.
This means there exist a positive constant C and C_0
Such that $t_1(n)+t_2(n) \le C$.

$$t_1(n) \le C_1 g_1(n) \text{ For all } n \ge n_1$$
$$t_2(n) \le C_2 \cdot g(n) \text{ For all } n \ge n_2$$

let $n_0 = \max\{n_1, n_2\}$ for all $n \ge n_0$
cosider $t_1(n)+t_2(n)$ for all $n \ge n_0$

$$t_1(n)+t_2(n) \le C_1 g_1(n) + C_2 \cdot g_2(n)$$

We need to Relate $g_1(n)$ and $g_2(n)$ to $\max\{g_1(n), g_2(n)\}$

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

Thus,

$$C_1 \cdot g_1(n) \le C_1 \max\{g_1(n), g_2(n)\}.$$
$$C_2 \cdot g_2(n) \le C_2 \cdot \max\{g_1(n), g_2(n)\}$$

$$C_1 g_1(n) + C_2 g_2(n) \le C_1 \cdot \max\{g_1(n), g_2(n)\} + C_2 \cdot \max\{g_1(n), g_2(n)\}$$
$$C_1 \cdot g_1(n) + C_2 g_2(n) \le (C_1+C_2) \max\{g_1(n), g_2(n)\}.$$
$$t_1(n)+t_2(n) \le (C_1+C_2) \max\{g_1(n), g_2(n)\} \text{ For all } n_1 \ge n_0$$

By the defination of Big O Notation.

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

Thus the assertion is proved.

2. Find the time Complexity of the Reccurce equation.

A: let us consider such that Reccuirence for merge sort.

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

By using master therom.

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

where $a \geq 1, b \geq 1$ and $F(n)$ is positive Function.

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

$a = 2, b = 2, F(n) = n$

By comparing of $F(n)$ with $n \log_b a$

$\log_b a = \log_2 2 = 1$

Compare $F(n)$ with $n.\log_b a$:

$F(n) = n$

$n.\log_b a = n^1 = n$

* $F(n) = O(n^{\log_b a})$, Then $T(n) = O(n^{\log_b a} \log n)$

In our Case:

$\log_b a = 1$

$T(n) = O(n^1 \log n) = O(n \log n)$

Time complexity of reccurence Relation is $T(n)$

$T(n) = 2T(n/2) + n$ is $O(n \log n)$

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

A: By Apply of Master theorem.

$T(n) = aT(n/b) + F(n)$ where $a \geq 1$, $b \geq 1$

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

Here $a = 2, b = 2, F(n) = 1$

By comparision of $F(n)$ and $n.\log_b a$.

If $F(n) = O(n^c)$ where $c < \log_b a$, Then $T(n) = O(n^{\log_b a})$

If $F(n) = O(n^{\log_b a})$, Then $T(n) = O(n^{\log_b a} . \log n)$

If $F(n) = \Omega(n^c)$ where $c > \log_b a$ Then $T(n) = O(F(n))$

lets calculate $\log_b a$:

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

$$F(n) = 1$$

$$n \cdot \log_b a = n^1 = n.$$

$F(n) = O(n^c)$ with $c < \log_b a$ (case 1)

In this case $c = 0$ and $\log_b a = 1$

$c < 1$ so, $F(n) = O(n^{\log_b a}) = O(n^1) = O(n)$.

Time complexity of Reccurence Relation.

$$T(n) = 2T(n/2) + 1 \text{ is } O(n)$$

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

A: Here where $n=0$

$$T(0) = 1$$

Reccurence relation Analysis.

For $n > 0$

$$T(n) = 2T(n-1)$$
$$T(n) = 2T(n-1)$$
$$T(n-1) = 2T(n-2)$$
$$T(n-2) = 2T(n-3)$$
$$T(1) = 2T(0).$$

From this pattern.

$$T(n) = 2 \cdot 2 \cdot 2 \cdots 2$$
$$T(0) = 2^n T(0)$$

Since $T(0) = 1$, We have.

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

The recurce Relation is $T(n) = 2T(n-1)$ for $n > 0$ and $T(0) = 1$ is $T(n) = 2^n$.

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

A: $F(n) = O(g(n))$ Means $C>0$ and $n_0 \geq 0$

$F(n) \leq C \cdot g(n)$ For all $n \geq n_0$.

Given is $F(n) = n^2+3n+5$

$C>0, n_0 \geq 0$ Such that $F(n) \leq C \cdot n^2$

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

let choose $C=2$.

$F(n) \leq 2 \cdot n^2$

$F(n) = n^2+3n+5 \leq n^2+3n^2+5n^2 = 9n^2$

So, $C=9$, $n_0=1$

$F(n) \leq 9n^2$ for all $n \geq 1$

$F(n) = n^2+3n+5$ is $O(n^2)$