Assignment - 10

NAME : G. Varun Kumar

Reg NO : 19237204D

Sub - code : CSA0670

2) 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 the assertions

Solution:

By definition, there exist constants c_1, n_1 such that for all $n \geq n_1$: $t_1(n) \leq c_1 \cdot g_1(n)$

similarly there exist constants c_2, n_2 such that for all $n \geq n_2$:

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

let $n_0 = \max(n_1, n_2)$ and $c = c_1 + c_2$. for all $n \geq n_0$: $t_1(n) + t_2(n) \leq c_1 \cdot g_1(n) + c_2 \cdot g_2(n)$

By definition of maximum:

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

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

Thus

$t_1(n) + t_2(n) \leq c_1 \max\{g_1(n), g_2(n)\} + c_2 \cdot \max\{g_1(n), g_2(n)\}$

$t_1(n) + t_2(n) \leq (c_1 + c_2) \max\{g_1(n), g_2(n)\}$

Hence

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

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

Solution:

To show $f(n) = n^2 + 3n + 5$ is $O(n^2)$:

$n^2 + 3n + 5 \leq c \cdot n^2$ $\quad f(n) = n^2 + 3n + 5$, $g(n) = cn^2$

for $c = 2$ and $n_0 = 3$:

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

for all $n \geq 3$

$\therefore f(n) = n^2 + 3n + 5$ is $O(n^2)$

2) Find the time complexity of the below recurrence equation

$T(n) = 3T(n/3) + n$

Solution: given $T(n) = 3T(n/3) + n$

1) identify the form: The recurrence relation is of the form

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

$a = 3, b = 3, f(n) = n$

2) Apply master theorem:

$T(n) = aT(n/b) + O(n^d)$:

Compute $\log_b a$:

$\log_{(3)} 3 = 1$

$d = 1$

$\log_b(a) = 1$

3) Determine the time complexity according to the master theorem

$d = \log_b a$:

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

$\therefore$ The time complexity of the recurrence relation is

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

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

Solution:

By applying of masters theorem

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

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

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

By comparison of $f(n)$ $n^{\log_b a}$

Let's calculate $\log_b a$:

$\log_b a = \log_2 2 = 1$

$f(n) = 1$

$n^{\log_b a} = n^1 = n$

$f(n) = O(n^c)$ with $c < \log_b a$

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

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

The complexity of recurrence relation.

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

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

Solution:

Here, where $n=0$

$T(0) = 1$

Recurrence 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

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

since $T(0) = 1$, we have $T(n) = 2^n$

The recurrence relation is

$T(n) = 2T(n-1)$ for $n > 0$ and $T(0) = 1$ is

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