1) If $t_1(n) \in O(g(n))$ 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.

Sol we need to show that $t_1(n)+t_2(n)$ to $\max\{g_1(n), g_2(n)\}$ this means three exist a positive constant c and n_0 such that

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

$t_1(n) \leq c g_1(n) \forall n \geq n_0$

$t_2(n) \leq c_2 g_2(n) \forall n_1 \geq n_0$

let $n_0 = \max\{n_1, n_2\} \forall n \geq n_0$

consider $t_1(n) + t_2(n) \forall n \geq n_0$

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

wee need to relate $g_1(n)$ and $g_2(n)$ to $\max\{g(n), g_2(n)\}$

$C_1 g_1(n) \leq C_1 \max\{g_1(n), g_2(n)\}$

$C_2 g_2(n) \leq C_2 \max\{g_2(n), g_2(n)\}$

$t_1(n) + t_2(n) \leq (C_1 + C_2) \max\{g_1(n), g_2(n)\}$ for all $n \geq n_0$

By the adefination of Bigo notation

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

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

thus the assertion is proved.

2) Find the time complexity of the recurrence relation.

let us consider such that recurrence for merge sort

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

By using master's theorem

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

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 $T(n) = O(n \log_b^a \cdot \log n)$

In our case

$\log_b^a = 1$

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

Then Time complexity of Recurrence relation is

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

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

By appling of master's theorem

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

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

Here $a=2, b=2$ $fn=1$

By comparing 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 $T(n) = O(n^{\log_b^a} \cdot \log n)$

If $f(n) = \Omega(n^c)$ wher $c > \log_b^a$ then $T(n) = O\ f(n)$

let 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 $c < \log_b^a$ (case 1)

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

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

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

Here, where $n=0$

$T(0) = 1$

Recurrence relation analysis

for $n>0$

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

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

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

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

$T(1) = 2T(0)$

for this partition

$T(n) = 2.2.\ldots\ T(0) = 2^n\ T(0)$

Since $T(0)=1$ we have

the Recurrence 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 $O(n^2)$

$f(n) = O(g(n))$ means $C>0$ and $n_0 \geq 0$

$f(n) \leq C.g(n)\ \forall\ n \geq n_0$

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

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

let Choose $C=2$

$f(n) \leq 2.n^2$

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

so $C=9,\ n_0=1,\ f(n) \leq 9n^2\ \forall\ n \geq 1$

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