Tutorial - 2

Q.1) $T(n) = \begin{cases} 7\,T\left(\frac{n}{2}\right) + n^2 & \text{if } n > 2 \\ 1 & \text{if } n \leq 2 \end{cases}$

$\longrightarrow$ $T(n) = 7T\left(\frac{n}{2}\right) + n^2$

$= 7\left[7T\left(\frac{n}{2^2}\right) + \left(\frac{n}{2}\right)^2\right] + n^2$

$= 7^2 T\left(\frac{n}{2^2}\right) + 7\left(\frac{n}{2}\right)^2 + n^2$

$= 7^2\left[7T\left(\frac{n}{2^3}\right) + \left(\frac{n}{2^2}\right)^2\right] + 7\left(\frac{n}{2}\right)^2 + n^2$

$= 7^3 T\left(\frac{n}{2^3}\right) + 7^2\left(\frac{n}{2^2}\right)^2 + 7\left(\frac{n}{2}\right)^2 + n^2$

$\vdots$

$= 7^k T\left(\frac{n}{2^k}\right) + 7^{k-1}\left(\frac{n}{2^{k-1}}\right)^2 + \cdots + 7^2\left(\frac{n}{2^2}\right)^2 + 7\left(\frac{n}{2}\right)^2 + n^2$

$= 7^k T\left(\frac{n}{2^k}\right) + \sum_{i=1}^{k-1} 7^i \left(\frac{n}{2^i}\right)^2 + n^2$

as $\frac{n}{2^k} = 1 \therefore n = 2^k$

$\therefore k = \log_2 n$

$\therefore = 7^{\log_2 n}(1) + \sum_{i=1}^{\log_2 n - 1} 7^i \left(\frac{n}{2^i}\right)^2 + n^2$

$= 7^{\log_2 n} + n^2$

$= n^{\log_2 7} + n^2$

$\Rightarrow O\left(n^{\log_2 7}\right)$

Q.2) $T(n) = \begin{cases} T(1) & \text{if } n=0 \\ 2T\left(\frac{n}{2}\right)+n & \text{if } n>1 \end{cases}$

$\rightarrow \quad T(n) = 2T\left(\frac{n}{2}\right)+n$

$= 2\left[2T\left(\frac{n}{2^2}\right)+\frac{n}{2}\right]+n$

$= 2^2 T\left(\frac{n}{2^2}\right)+2\frac{n}{2}+n$

$= 2^2 T\left(\frac{n}{2^2}\right)+2n$

$= 2^2\left[2T\left(\frac{n}{2^3}\right)+\frac{n}{2^2}\right]+2n$

$= 2^3 T\left(\frac{n}{2^3}\right)+2^2\cdot\frac{n}{2^2}+2n$

$= 2^3 T\left(\frac{n}{2^3}\right)+3n$

$\vdots$

$= 2^k T\left(\frac{n}{2^k}\right)+k\cdot n$

as $\frac{2^k}{n}=1 \quad \therefore 2^k=n$

$\therefore k=\log_2 n$

$= n\cdot(1)+\log_2 n\times n$

$= n\log n+n$

$$\boxed{T(n)=O(n\log n)}$$

Q.3) $T(n) = \begin{cases} a & n=0 \\ 2T(n/2)+bn & n>1 \end{cases}$

$\longrightarrow$ $T(n) = 2T\left(\frac{n}{2}\right) + bn$

$= 2\left[2T\left(\frac{n}{2^2}\right) + b\left(\frac{n}{2}\right)\right] + bn$

$= 2^2 T\left(\frac{n}{2^2}\right) + bn + bn$

$= 2^2 T\left(\frac{n}{2}\right) + 2bn$

$= 2^2\left[2T\left(\frac{n}{2^3}\right) + b\frac{n}{2^2}\right] + 2bn$

$= 2^3 T\left(\frac{n}{2^3}\right) + bn + 2bn$

$= 2^3 T\left(\frac{n}{2^3}\right) + 3bn$

$\vdots$

$= 2^k T\left(\frac{n}{2^k}\right) + k.bn$

as $2^k = n$
$k = \log_2 n$

$\therefore = n\cdot(1) + \log_2 n . bn.$

$\therefore T(n) = O(n\log_2 n)$

Page No. Date

Q.4) $T(n) = \begin{cases} c & n = 1 \\ 2T(n/2) + c & n > 1 \end{cases}$

$\longrightarrow$ $T(n) = 2T\left(\frac{n}{2}\right) + c$

$= 2\left[2T\left(\frac{n}{2^2}\right) + c\right] + c$

$= 2^2 T\left(\frac{n}{2^2}\right) + 2c + c$

$= 2^2\left[2T\left(\frac{n}{2^3}\right) + c\right] + 2c + c$

$= 2^3 T\left(\frac{n}{2^3}\right) + 2^2 c + 2^1 c + 2^0 c$

$\vdots$

$= 2^k T\left(\frac{n}{2^k}\right) + 2^{k-1} c + \dots + 2^2 c + 2^1 c + 2^0 c$

$= 2^k T\left(\frac{n}{2^k}\right) + (2^k - 1)c$

as $2^k = n$

$k = \log_2 n$

$= n \cdot (1) + (2^{\log_2 n} - 1)c$

$= O(n)$

$\boxed{T(n) = O(n)}$

Q.5) recursive algorithm of factorial and its recurrence relation.

→

```
fact (n)                          → T(n)
{
  if n==0 || n==1 then            → 1
      return 1;
  else
      return fact(n-1) * n;       → T(n-1)
}
```

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

$\therefore$ $T(n) = T(n-1) + 1$

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

$= T(n-2) + 2$

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

$= T(n-3) + 3$

$\vdots$

$= T(n-k) + k.$

$n - k = 0$

$n = k$

$\therefore$ $T(n) = T(0) + n$

$T(n) = n$

$$\boxed{T(n) = O(n)}$$

Q.6) recursive algorithm of fibonacci series and solve recurrence relation.

$\longrightarrow$

```
fibo(n)                                      T(n)
{
   if n <= 1                                 —— 1
    return n;
  else
   return fibo(n-1) + fibo(n-2);      —— T(n-1) + T(n-2)
}
```

$\therefore \; T(n) = T(n-1) + T(n-2) + c$

Assume $T(n-1) \approx T(n-2)$

$\therefore \; T(n) = 2T(n-2) + c$

$= 2\left[2T(n-4) + c\right] + c$

$= 4T(n-4) + 2c + c$

$= 4\left[2T(n-6) + c\right] + 2c + c$

$= 8T(n-6) + 4c + 2c + c$

$= 2^3 T(n-6) + 2^2 c + 2^1 c + 2^0 c$

$\vdots$

$= 2^k T(n-2k) + (2^k - 1)c$

$\because \; n - 2k = 0 \;\therefore\; n = 2k$

$k = n/2$

$T(n) = 2^{n/2} \cdot T(0) + (2^{n/2} - 1)c$

$T(n) \propto 2^{n/2}$ (lower bound)

now Assume,

$T(n-2) \approx T(n-1)$

$\therefore T(n) = 2T(n-1) + c$

$= 2[2T(n-2) + c] + c$

$= 2^2 T(n-2) + 2c + c$

$= 2^2 [2T(n-3) + c] + 2c + c$

$= 2^3 T(n-3) + 2^2 c + 2^1 c + 2^0 c$

$\vdots$

$= 2^k T(n-k) + (2^k - 1)c$

$\because n-k = 0 \quad \therefore k = n$

$= 2^n \cdot T(0) + (2^n - 1)c$

$T(n) \propto 2^n$ (upper bound)

$\therefore \boxed{T(n) = O(2^n)}$

Page No.
Date

Q.7) show that the following equalities are correct

a) $5n^2 - 6n \Rightarrow \Theta(n^2)$

$f(n) = 5n^2 - 6n$
for time complexity we take n term with highest degree and neglect coefficient.
so,

$\underline{\underline{\Theta(n^2)}}$

~~b) $n! \Rightarrow O(n^n)$~~

b) $n^3 + 10^6 n^2 \Rightarrow \Theta(n^3)$

$f(n) = n^3 + 10^6(n^2)$
here we take highest degree term of n i.e. n^3
so,

$\underline{\underline{\Theta(n^3)}}$

Q.8) Show that the following equalities are incorrect:

a) $10n^2 + 9 \Rightarrow O(n)$
$f(n) = 10n^2 + 9$
here for time complexity we select highest degree term of n and coefficient of that term neglected so,
$\Rightarrow O(n^2)$
so $O(n)$ is ~~o~~ incorrect.

b) $n^2 \log n \Rightarrow O(n^2)$
$f(n) = n^2 \log n$
as here only one term ~~on~~ of n so
$O(n^2 \log n)$. so $O(n^2)$ is incorrect.

c) $\frac{n^2}{\log n} \Rightarrow O(n^2)$
$f(n) = \frac{n^2}{\log n}$

as here only one term of n so
$O(n^2/\log n)$ so $O(n^2)$ is incorrect.