Muhammad Nadeem
20198000$\bar{5}$0
Analysis of Algorithm
Section- A

# Assignment # 01

**QNo1:-** $T(n) = T(n/3) + cn$ , $T(n) = n$

**Solution:-**

General form of Master Theorem is

$$T(n) = a * T(n/b) + f(n)$$

Conditions:- $a \geq 1$ , $b > 1$

Here, $a = 1$ , $b = 3$ , $f(n) = cn$

General solution of Master Theorem will be:

$$T(n) = n^{\log_b a} [u(n)] \overset{①}{\Longrightarrow} \quad H(n) = \frac{f(n)}{n^{\log_b a}}$$

Cases:-

| | | if $R(n)$ | $u(n)$ |
|---|---|---|---|
| | Case 1 | $n^R$, if $r > 0$ | $O(n^2)$ |
| | Case 2:- | | |
| | Case 3:- | $n^R$, $R > 0$ | $O(1)$ |
| | | $(\log n)$, $i \geq 0$ | $\frac{(\log_2 n)^{i+1}}{i+1}$ |

By applying Case 2

$$h(n) = \frac{c}{n^{\log_3 1}} \Rightarrow n^1 \text{ and } u(n) = d(n)$$

Putting it in eq (1)

$$T(n) = n^{\log_3 1} * O(n)$$

$$= n^3 \, O(n) = 1 \times O(n)$$

$$= O(n)$$

Question 2

$T(n) = 8T(n/2) + qn$ if $n>1$

By Master $T(n) = a * T(n/b) + f(n)$

Here $a=8, b=2, f(n) = qn$

General solution is-

$$T(n) = n^{\log_b^a} [U(n)]$$

$$H(n) = \frac{f(n)}{n^{\log_b^a}}$$

$$= \frac{qn}{n^{\log_2^8}}$$

$$= \frac{qn}{n^3}$$

$$= q\,n^{-2}$$

$H(n) = n^{-2}$

Case 2:-

$n^{\gamma} = O(1)$

$n^{-2} = O(1)$

Putting value of $U(n)$

$T(n) = n^{\log_2^8} \times \Theta(1)$

$T(n) = n^3$

So, $T(n) = O(n^3)$

**Q no 3:-** $T(n) = 7T(n/2) + qn$ , if $n > 1$

Sol:- values of $a, b$ $f(n)$ using Master theorem general form:- $T(n) = aT(n/b) + f(n)$

$a = f$ , $b = 2$, $f(n) = qn$

General:-

$T(n) = n^{\log_b^a} [U(n)]$

$U(n) = H(n) = \frac{f(n)}{n^{\log_b^a}} \Rightarrow \frac{qn}{n^{\log_2^7}} \Rightarrow \frac{qn}{n^2} \Rightarrow q n^{-1}$

$H(n) = n^{-1}$

by Case 2:- $n^{-1} = O(1)$

by Putting

$T(n) = n^{\log_b^a} [U(n)]$

$= n^2 * O(1)$

$= O(n^2)$

**Q no 4:-** $T(n) = T(n/2) + \log n$ if $n > 1$

values by using Master theorem

$a = 1$, $b = 2$, $f(n) = \log n$

$U(n)$ ? $\Rightarrow H(n) = \frac{f(n)}{n^{\log_b^a}}$

$= \frac{\log^n}{n^0} = \frac{\log^n}{1} = \log n$

Case #3

$= \frac{(\log_2^n)^{i+1}}{i+1}$

$= \frac{(\log_2^n)^{1+1}}{1+1} = (\log_2^n)^2$

by putting it:-

$T(n) = a\ T(n/b) + f(n)$
$= 1\ T(n/2) + f(n)$
$T(n) = n^{\log_2 1} [u(n)]$
$= n^0 (\log_2 n)^2$
$= 1 * (\log_2 n)^2$
$T(n) = O(\log_2 n)^2$

---

**Q.No 5:-** $T(n) = T(n-1) + 1$

Logic base for input n, the algorithm by watching the value changing inside the given equation is T(n-1) by n

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

T(n) take $T(n-1) + \boxed{1}$

then T(n-2) will be

$T(n-3) + 2$

So,

$T(n) = 1 + T(n-1)$
$T(n-1) = 2 + T(n-2)$
$T(n-2) = 3 + T(n-3)$
$\vdots \qquad \vdots$

$T(n-k) = k + T(n-k)$ It stop when input is 0.

$n-k=0$
$n=k$ Here is Algo stop.

So, $T(n-k+1) = n + T(n-n)$

$= n + T(n-n)$

$= n + T(0)$

$O(n)$.

---

**Question 5:- Remaining substitution:-**

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

for $T(n-1)$

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

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

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

$= T(n-3)+3$

So, if we continue this

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

As, we know that

$n-k=0$

$n=k$ Algo stop here

Hence, $T(n) = T(n-n)+n$

$= T(0)+n$

$T(n) = 1+n$

So,

$O(n)$.

---

**Q No 6:-** $T(n) = 3T(n/3)+n/2$, if $n>1$

by Master theorem:-

$T(n) = a\,T(n/b)+f(n)$

$a=3$, $b=3$; $f(n) = n/2$

$$h(n) = \frac{f(n)}{n^{\log_b a}}$$

$$= \frac{n/2}{n^{\log_3 3}} = \frac{n/2}{n/1} = \frac{n}{2}\times\frac{1}{n} = \frac{1}{2}$$ It's a Constant.

Case 3:-

$$\frac{(\log_2^n)^{0+1}}{0+1}$$

$\log_2^n$ General Sol:-

$T(n) = n^{\log_b^a} [U(n)]$

$= n^{\log_3^3} [\log_2^n]$

$= n \cancel{*} \log_2^n$

$= O(n \log_2^n)$

---

**Q no 7:-** $T(n) = 3T(n/3) + n\log n$

by Master theorem

$T(n) = a\,T(n/b) + f(n)$

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

Sol: $T(n) = n^{\log_b^a} [U(n)]$

$U(n) = H(n)$

$$H(n) = \frac{f(n)}{n^{\log_b a}}$$

$$= \frac{n\log n}{n^{\log_3^3}}$$

$= \log n$

Case 3:-

$$= \frac{(\log_2^n)^{1+1}}{1+1}$$

$$= \frac{(\log_2^n)^2}{2}$$

$= (\log_2^n)^2$

Putting in eq

$T(n) = n^{\log_b^a} (\log_2^n)^2$

$= n^{\log_3^3} (\log_2^n)^2$

$T(n) = n(\log_2^n)^2$

$O[n(\log_2^n)^2]$