Recurrences → A recurrences is an eqn in its relation or inequality

How to Solve a Recursive Problem

Recursion is a method which calls a function ~~repeatedly~~ to itself until a base condition or terminating condition doesn't reach.

Recurrence Relation

Ln (Want of find)

$$T(n) = \begin{cases} 1, & \text{if } n==0 \,||\, n==1 \\ \cancel{n * T(n-1), \text{ if } n>1} & \\ n * fact(n-1), & \text{if } n>1 \end{cases}$$

Time complexity of the same can be evaluated as:

$$T(n) = \begin{cases} 1 & \text{if } n==0 \,||\, n==1. \\ n*T(n-1), & \text{if } n>1. \end{cases}$$

There are four methods are used to find the time complexity of a given function or of a problem:

(i) Substitution Method → Guess the solution → Use mathematical induction to solve boundry condition

(ii) Iterative method

(iii) Recursive Tree Method

(iv) Master Method

Substitution Method: In general substitution method gives a best solution of any of the problem, whereas others method may be best suit for other problems, or may specific problems.

Advantege Every problem (recursive problem) can be solved using substitution problem.

Disadvantage Mathematical calculation is little enough viewed to others.

Recurrences → A recurrences is an eqn in its relation or inequality

classmate
Date
Page

How to Solve a Recursive Problem

Recursion is a method which calls a function ~~repeatedly~~ to itself until a base condition or terminating condition doesn't reach.

Recurrence Relation

Lx (Want of find)

$$T(n) = \begin{cases} 1, & \text{if } n==0 \,||\, n==1 \\ \cancel{n * T(n-1), \text{ if } n>1} & \\ n * fact(n-1), & \text{if } n>1 \end{cases}$$

Time complexity of the same can be evaluated as:

$$T(n) = \begin{cases} 1 & \text{if } n==0 \,||\, n==1. \\ n * T(n-1), & \text{if } n>1. \end{cases}$$

There are four methods are used to find the time complexity of a given function or of a problem:

(i) Substitution Method → Guess the solution; → Use mathematical induction to solve boundry condition
(ii) Iterative method
(iii) Recursive ~~Tr~~ Tree Method
(iv) Master Method

Substitution Method: In general substitution method gives a best solution of any of the problem, whereas others method may be best suit for other problems, or may specific problems.

Advantage Every problem (recursive problem) can be solved using ~~a~~ substitution problem.

Disadvantage Mathematical calculation is little enough compared to others.

Problem-1

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \text{ Base condition} \\ n * T(n-1), & \text{if } n > 1 \end{cases}$$

classmate
Date
Page

$T(n) = n * T(\underline{n-1})$ ——— (1)

→ This function is decreasing by 1

$T(n-1) = (n-1)\, T(n-1-1)$

$= (n-1)\, T(n-2)$ ——— (2)

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

$= (n-2) * T(n-3)$ ——— (3)

Here we are finding the iteration & no need to compute more than 3 ~~ok~~ times.

Now

Substitute $T(n-1)$ from eqn (2) in (1)

$T(n) = n * (n-1)\, T(n-2)$ ——— (4)

~~Substitute $T(n-2)$ from eqn 3 in (4)~~

~~$T(n) = n * (n-1) * (n-2)\, T(n-2-1)$~~

Here we ~~can~~ get a trend how the function is decreasing

$T(n) = n * (n-1) * (n-2)\, T(n-3)$

(n-1) steps

Now we need to eliminate by terminating condition.

$T(n) = n * (n-1) * (n-2) * (n-3) * \cdots * 1$

$= n * (n-1) * (n-2) * (n-3) * \cdots 3 * 2 * 1$

$= n * n\left(1 - \frac{1}{n}\right) * n\left(1 - \frac{2}{n}\right) * n\left(1 - \frac{3}{n}\right) * \cdots * n\left(\frac{3}{n}\right) * n * \left(\frac{2}{n}\right) * n$

→ max of this will always less than of n

$T(n) = O(n^n)$.

Problem - 2

Suppose we have a problem (recursive)

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

$$T(n) = 2T\left(\frac{n}{2}\right) + n \quad \text{—— (1)}$$

$$T\left(\frac{n}{2}\right) = 2\left(T\left(\frac{1}{2}\cdot\frac{n}{2}\right) + \frac{n}{2}\right) + n$$

$$= 2T\left(\frac{n}{4}\right) + \cancel{\tfrac{n}{2}}\, 2n \quad \text{—— (2)}$$

$$T\left(\frac{n}{4}\right) = 2\left(T\left(\frac{1}{2}\cdot\frac{n}{4}\right) + \frac{n}{4}\right) \quad \text{—— (3)}$$

Now, substitute (2) in (1)

$$T(n) = 2\left(2T\left(\frac{n}{4}\right) + \frac{n}{2}\right) + n$$

$$= 4T\left(\frac{n}{4}\right) + n + n$$

$$= 4T\left(\frac{n}{4}\right) + 2n \quad \text{—— (4)}$$

Now, substitute (3) in (4)

$$T(n) = 4\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\right) + 2n$$

$$= 8T\left(\frac{n}{8}\right) + n + 2n$$

$$= 8T\left(\frac{n}{8}\right) + 3n \quad \text{—— (5)}$$

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

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

$\log_2 n = K$

$T(n) = n + \log_2 n \cdot n$

$= n + n\cdot\log_2 n$

$= n\log_2 n$

$T(n) =$

$$\boxed{T(n) = O(n \log_2 n)}$$

Here we have ~~to assume~~, $n = 2^K$, K no of steps so that ~~that~~ $T\left(\frac{n}{2^K}\right) = T\left(\frac{n}{n}\right) = T(1)$ ~~to~~ should ~~to~~ come.

$T(n) = 2^K \cdot T(1) + \underline{K}\cdot n$. Now represent K in n by taking a log to both sides

Problem-3 For the recurrence relation $T(n) = T\left(\frac{n}{2}\right) + 1$, show that it is asymptotically bounded by $O(\log n)$.

Soln:

$$T(n) = T\left(\frac{n}{2}\right) + 1 \quad \text{--- (1)}$$

$$T\left(\frac{n}{2}\right) = T\left(\frac{1}{2} \cdot \frac{n}{2}\right) + 1$$

$$= T\left(\frac{n}{4}\right) + 1 \quad \text{--- (2)}$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{1}{2} \cdot \frac{n}{4}\right) + 1 \quad \text{---}$$

$$= T\left(\frac{n}{8}\right) + 1 \quad \text{--- (3)}$$

$$\Rightarrow T\left(\frac{n}{2^3}\right) + 1 \quad \text{---}$$

Now

Substitute eqn. (2) in eqn (1)

$$T(n) = T\left(\frac{n}{4}\right) + 1 + 1$$

$$= T\left(\frac{n}{4}\right) + 2 \quad \text{--- (4)}$$

Substitute eqn (3) in eqn (4)

$$T(n) = T\left(\frac{n}{8}\right) + 1 + 2$$

$$= T\left(\frac{n}{2^3}\right) + 3 \quad \text{--- (5)}$$

Now, assume we are eqn. (5) ~~total~~ to kth steps is equal to $n = 2^k$, $\log_2 n = \log_2(2^k) \Rightarrow k = \log_2 n$

$$T(n) = T(1) + \log_2 n$$

classmate
Date
Page

Binary Search

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

$$T(n) = T\left(\frac{n}{2}\right) + C \quad \text{— } 1$$

$$T\left(\frac{n}{2}\right) = T\left(\frac{1}{2}\cdot\frac{n}{2}\right) + C$$

$$= T\left(\frac{n}{4}\right) + C \quad \text{— } 2$$

$$T\left(\frac{n}{4}\right) = T\left(\frac{n}{8}\right) + C \quad \text{— } 3$$

~~Subs~~ eqn 2 in eqn. 1

$$\cancel{T(\tfrac{n}{2}} \; T(n) = T\left(\frac{n}{4}\right) + C + C$$

$$= T\left(\frac{n}{2^2}\right) + 2C \quad \text{— } 4$$

eqn 3 in eqn 4 substitute

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

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

Next trend is

$$T(n) = T\left(\frac{n}{2^4}\right) + 4C$$

$$T(n) = T\left(\frac{n}{2^{\cancel{k}5}}\right) + 5C$$

k steps & Eliminate it

How this will come, due it if we reach $n = 2^k$

$$\cancel{T(n) = T(\tfrac{n}{4}\ldots} \; T(n) = T\left(\frac{n}{2^k}\right) + kC$$

Suppose ~~if~~ $n = 2^k$, then $T\left(\frac{n}{2^k}\right) = T(1)$ & take log to both sides

$$\log n = \log_2(2^k)$$

$$\log_2 n = k$$

classmate Date Page

Iterative Recurrence Relation

Solving reccurrence with Iterative Method

"Unfold" the recurrence until you see the pattern to find. ~~Theorem~~

$$T(n) = 8T\left(\frac{n}{2}\right) + n^2 \quad \text{and} \quad T(1) = 1$$

$$T(n) = n^2 + 8T\left(\frac{n}{2}\right) \quad \text{--- (1)}$$

$$T\left(\frac{n}{2}\right) = \left(\frac{n}{2}\right)^2 + 8T\left(\frac{n}{2}\cdot\frac{1}{2}\right) \quad \text{--- (2)}$$

Put eqn (2) in eqn 1

$$T(n) = n^2 + 8\left[\frac{n^2}{4} + 8T\left(\frac{n}{4}\right)\right]$$

$$= n^2 + 2n^2 + 8^2\, T\left(\frac{n}{4}\right) \quad \text{--- (3)}$$

Now, for $T(n/4)$

$$T\left(\frac{n}{4}\right) = \left(\frac{n}{4}\right)^2 + 8T\left(\frac{n}{4}\cdot\frac{1}{2}\right)$$

$$= \frac{n^2}{\cancel{16}\,4^2} + 8T\left(\frac{n}{8}\right) \quad \text{--- (4)}$$

Put eqn (4) in eqn (3)

$$T(n) = n^2 + 2n^2 + 8^2\left[\frac{n^2}{4^2} + 8T\left(\frac{n}{8}\right)\right]$$

$$= n^2 + 2n^2 + 2^2 n^2 + 8^3\, T\left(\frac{n}{2^3}\right)$$

We keep expansion till ~~kth term~~ k^{th} term.

$$n = 2^k, \quad \log_2 n = \log_2 2^k = k$$

$$T(n) = n^2 + 2^1 n^2 + 2^2 n^2 + 2^3 n^2 + \dots + 2^{\log_2 n} n^2 + 8^{\log_2 n}$$

$$= n^2\underbrace{\left[1 + 2^1 + 2^2 + 2^3 + \dots + 2^{\log_2 n}\right]}_{\text{This is a Geometric Progression}} + 8^{\log_2 n}$$

$$T(n) = n^2 \sum_{K=0}^{\log_2 n-1} 2^K + (2^3)^{\log_2 n}$$

$$= n^2\, \theta\left(2^{\log_2 n-1}\right) + n^3$$

$$T(n) = \theta(n^3)$$

A geometric progression is a form of $a + ar + ar^2 + ar^3 \ldots\ldots$

In Iteration method, the recurrence is expanded as summation of terms, then summation provides the solution

Example-1 For recurrence relation $T(n) = T(n-1)+1$ and $T(1) = \theta(1)$, show the relation is asymptotically bounded by $\theta(n)$.

Sol^n:- Given that $T(n) = T(n-1)+1$

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

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

Also, $T(n-2) = T(n-2-1)+1$
$$= T(n-3)+1$$

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

From the above we can conclude that

$T(n) = T(n-K)+K$, where $K = n-1$

so, $T(n) = T(n-\overline{n-1}) + (n-1)$
$$= T(1) + n-1$$
$$= \theta(1) + n-1$$
$$= \theta(n)$$

Thus $T(n) = \theta(n)$.

OR

The same problem can be like this also

$$T(n) = T(n-1) + 1 \quad \text{——— (1)}$$

$$T(n) = 1 + T(n-1) \quad \text{——— (2)}$$

$$T(n) = 1 + 1 + T(n-2) \quad \text{Put (2) rec. } n-1 \text{ in (1)}$$

$$= 1 + 1 + 1 + T(n-3) = \underline{3} + T(n-3)$$

It termination will be at k^{th} term
$K = n-1$

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

$$= 1 + 1 + 1 + \cdots + T(2) + T(1)$$

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

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

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

Thus, $T(n) = \theta(n)$.

Example – 2

Solve the following recurrence relation using Iteration method

$$T(n) = 3T(\lceil n/4 \rceil) + n$$

$$T(n) = n + 3T(n/4)$$

$$= n + 3\left[\frac{n}{4} + 3T\left(\frac{n}{4}\cdot\frac{1}{4}\right)\right]$$

$$= n + \frac{3n}{4} + 9T(n/16)$$

$$= n + \frac{3n}{4} + 9\left[\frac{n}{16} + 3T\left(\frac{n}{16}\cdot\frac{1}{4}\right)\right]$$

Ceiling takes the real number x, and gives the least integer greater than or equal to x.
Example
ceil(2.4) = 3
$\lceil 2 \rceil = 2$, $\lceil 10.2 \rceil = 10$

$$T(n) = n + \frac{3n}{4} + 3^2\frac{n}{4^2} + 3^3 T\left(\frac{n}{4^3}\right)$$

$$= n + 3^1\frac{n}{4^1} + 3^2\frac{n}{4^2} + 3^3 T\left(\frac{n}{4^3}\right)$$

Suppose $n = 4^k$

$\log_2 n = \log_2 4^k$

$\log_2 n = k \log_2 4$

$$K = \frac{\log_2 n}{\log_2 4}$$

$= \log_2 n - \log_2 4$

$= \log_2 n - \log_2 2^2$

$= \log_2 n - 2\log_2 2$

$= \log_2 n - 2$

~~Suppose~~ $n = \cancel{3^{n-1}} \cdot 4^k$

$\frac{n}{3^n \cdot 3}$

$\frac{3^n}{3^n} = 4^k$

Next

$3^4 T\left(\frac{n}{4^4}\right)$

$$T(n) = n + 3^1\frac{n}{4^1} + 3^2\cdot\frac{n}{4^2} + 3^3\frac{n}{4^3} + \dots$$

$$+ 3^k T\left(\frac{n}{4^k}\right)$$

The series will be terminate only if $\frac{n}{4^k} = 1$, as it is given $T(1) = 1$

i.e $n = 4^k$

$\log n = \log 4^k$

$K = \log_4 n$

$\log n = k \log 4$

$K = \frac{\log n}{\log 4} = \log_4 n$

The series becomes as

$$T(n) = n + \frac{3n}{4} + 3^2\frac{n}{4^2} + 3^3\frac{n}{4^3} +$$

$$---- + 3^k T\left(\frac{n}{4^k}\right)$$

$$T(n) \le n + \frac{3n}{4} + 3^2\frac{n}{4^2} + \cdots + 3^{\log_4 n} T(1)$$

$$\le n\left(1 + \frac{3}{4} + \frac{3^2}{4^2} + \cdots + 3^{\log_4 n}\right)$$

$$\le n\left(1 + \frac{3}{4} + \frac{3^2}{4^2} + \cdots + n^{\log_4 3}\right)$$

$$\le n\left(\frac{1}{1-3/4}\right) + n^{\log_4 3}$$

$$\le 4n + n^{0.75}$$

$$\le 4n + n$$

$$T(n) \le 5n \quad \text{ie } f(n) \le M|g(n)|$$

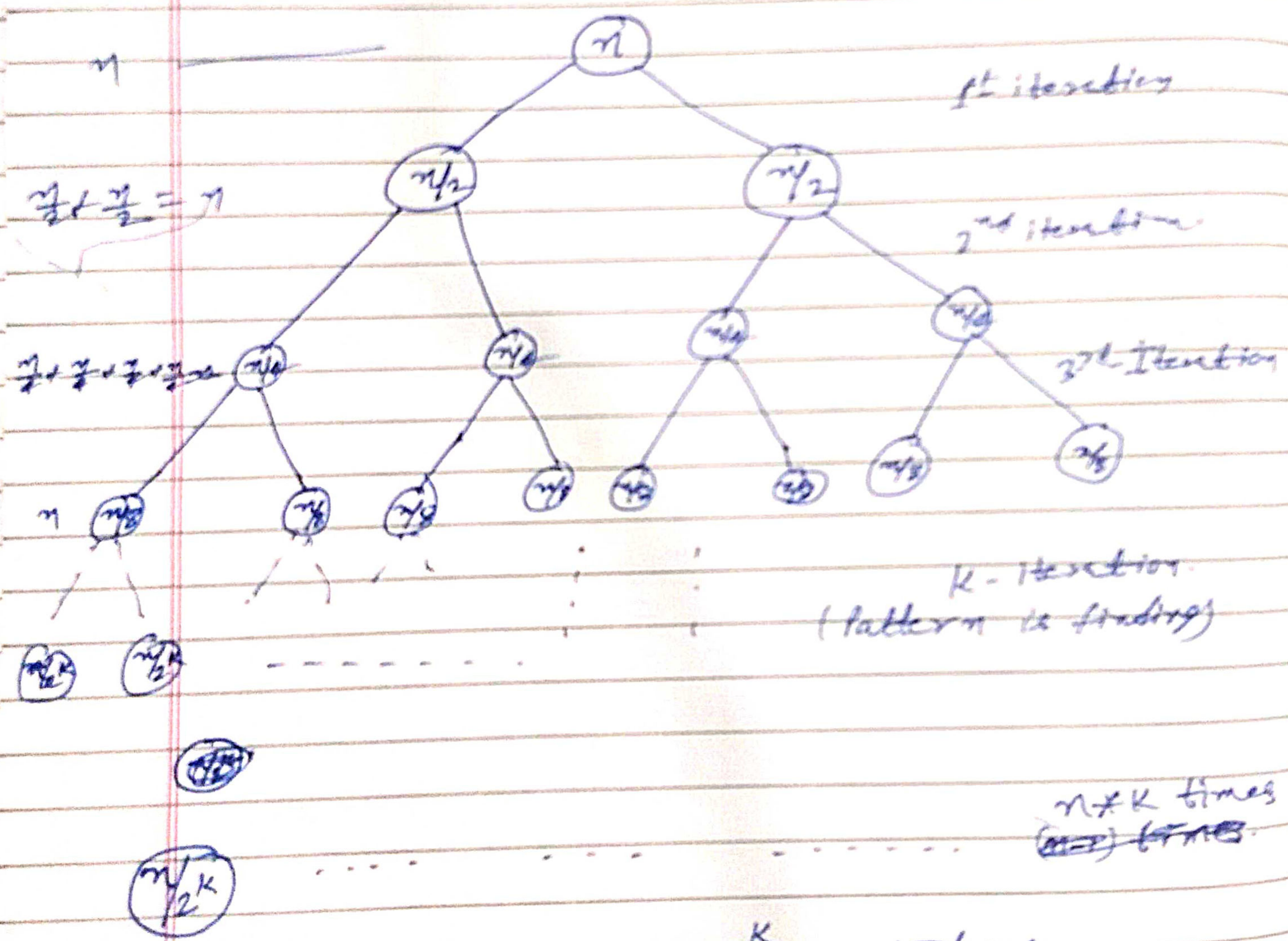
$$T(n) = O(n).$$

Recursive Tree Method

Recursive tree is a just a pictorial representation of iteration method and a solution of the problems where more than one recursive function present in recurrence relation problem.

Example:

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



Suppose $n = 2^k$

$\log n = \log 2^k = k$

~~T(n) = Ok~~

$n + n + n + \ldots$ k times $= n * k$

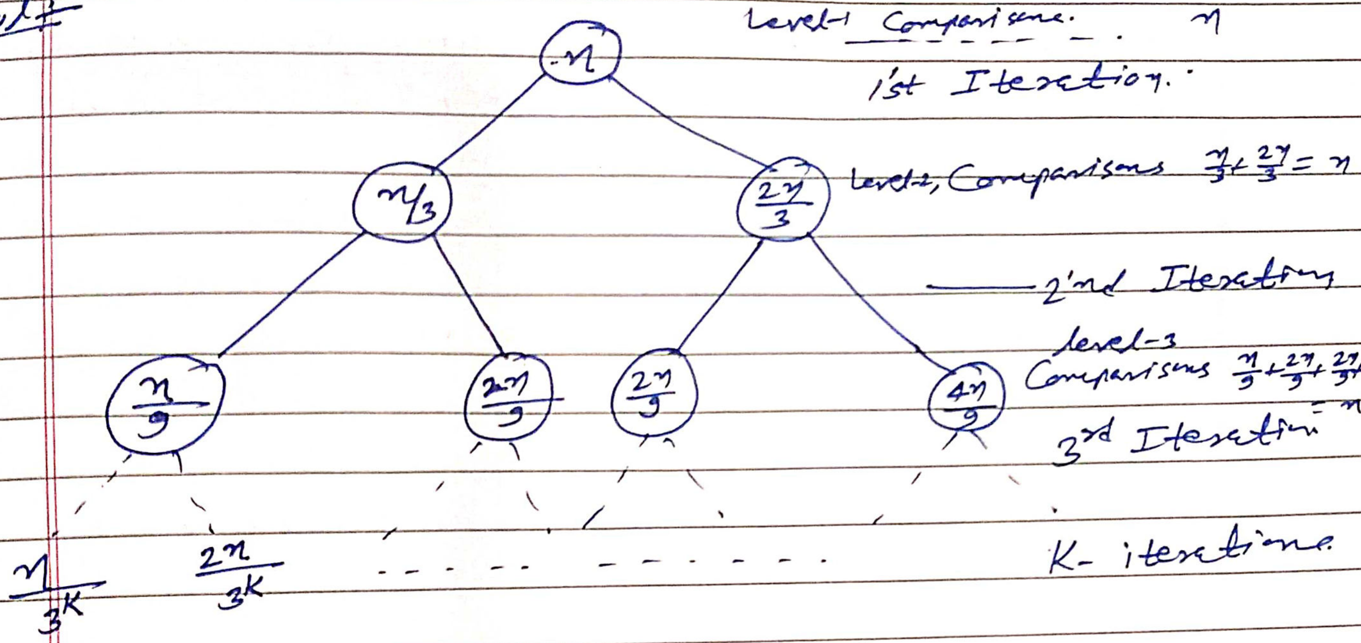
$$T(n) = O(n \cdot \log n)$$

Example-2 :- Consider the following recurrence

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

Obtain the asymptotic bound using recursion tree method.

Sol^n



~~When~~ $\frac{n}{3^K} + \frac{2n}{3^K} + \cdots + \frac{2^K n}{3^K}$ .. Patterns find.

When we add the values across the levels of the recursion tree, we get a value of n for every level. ~~The~~

We have to assume that

$$\left(\frac{2}{3}\right)^K \cdot n = 1, \quad \text{or} \quad n = \left(\frac{3}{2}\right)^K$$

$$\log_{3/2} n = \log_{3/2} {}^K = K$$

$$\boxed{K = \log_{3/2} n}$$

Thus the height of ~~the~~ ~~B~~ tree is

$= n + n + n + \cdots$ K. times.

$= n \cdot K$

$= n \cdot \log_{3/2} n$

$$T(n) = O(n \log_{3/2} n)$$

Master Method

We have studied about Substitution, Iterative & Recursive tree method. Any of the problem can be solved using above three methods, ~~and~~ but problem is that all methods needs a lot of calculation & consider to be slow method.

One of the Master method which can solve the problem in short calculations, but this problem is having some certain limitations. The problem must be in below format:

$$T(n) = aT\left(\frac{n}{b}\right) + f(n)$$

where $a \geq 1$ & $b > 1$

1. Suppose we have one problem

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

Here just check $a = 1$, $b = 1$, It is not equal to $b > 1$, so we can't apply master method, but we can apply other methods.

2. Suppose we have

$$T(n) = 8T(n/2) + n^2$$

$a = 1$, $b = 2$ Here we can apply master ~~theory~~ method. Then its solution will be

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

For ~~to~~ finding the $U(n)$, we have to find $H(n)$, where $H(n) = \frac{f(n)}{n^{\log_b a}}$

For remembrance

if $h(n)$	$U(n)$
① $n^r, r>0$	$O(n^r)$
② $n^r, r<0$	$O(1)$
③ $(\log n)^i, i \geq 0$	$\frac{(\log_2 n)^{i+1}}{i+1}$

~~Suppose~~ We consider

$$T(n) = 8T(n/2) + n^2$$

$$a = 8, \ b = 2, \ f(n) = n^2$$

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

$$= n^{\log_2 8} \ U(n)$$

$$= n^3 \cdot U(n) \quad \text{—— ①}$$

$$H(n) = \frac{f(n)}{n^{\log_b a}} = \frac{n^2}{n^{\log_2 8}} = \frac{n^2}{n^3} = \frac{1}{n} = n^{-1}$$

$n^{-1}, \ r<0$

$$T(n) = n^3 \cdot O(1) = O(n^3)$$

Example-3

$$T(n) = 9T(n/3) + n$$

$a = 9,\ b = 3,\ f(n) = n$

$$T(n) = n^{\log_b a} \cdot U(n)$$
$$= n^{\log_3 9} \cdot U(n) = n^2 \cdot U(n)$$

$U(n)$ depends upon $H(n) = \dfrac{f(n)}{n^{\log_b a}}$

$$= \frac{n}{n^{\log_3 9}} = \frac{n}{n^2} = \frac{1}{n} = n^{-1}$$

r is less than ~~of~~ 0, it means

$$H(n) = O(1) = U(n)$$

$$T(n) = n^2 \cdot O(1) = n^2$$

Example-4

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

$a = 1,\ b = \frac{3}{2},\ f(n) = 1$

$$T(n) = n^{\log_b a} \cdot U(n)$$
$$= n^{\log_{(3/2)} 1} \cdot U(n)$$
$$= n^0 \cdot U(n)$$

$$U(n) = H(n) = \frac{f(n)}{n^{\log_b a}} = \frac{1}{n^0}$$

$$= \frac{1}{n^{\log_b a}} = \frac{1}{n^{\log_{2/3} 1}} = 1$$

$= \xi \rightarrow f(n) \cdot n^{-\log_b a} = 1 \cdot n^{-\log_{(2/3)} 1}$

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

classmate
Date
Page

Now $T(n) = n^0 \cdot U(n) = 1 \cdot \log_2 n = O(\log_2 n)$

Example-5

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

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

Sol^n:

$$T(n) = n^{\log_b a} \cdot U(n)$$
$$= n^{\log_2 1} \cdot U(n) = n^0 \cdot U(n)$$
$$= U(n)$$

$$H(n) = \frac{f(n)}{n^{\log_b a}} = \frac{c}{n^{\log_2 1}} = \frac{c}{1} = c$$

We have to make a third case of $U(n)$ as

$$H(n) = (\log_2 n)^0 \cdot c$$
$$= \frac{(\log_2 n)^{0+1}}{0+1} = \log_2 n \cdot c$$
$$= O(\log_2 n)$$

Example-6

$$T(n) = 3T(n/4) + n\log n$$

$a=3,\ b=4,\ f(n) = n\log n$

$$T(n) = n^{\log_b a} \cdot U(n) = n^{\log_4 3} \cdot U(n)$$
$$= n^0 \cdot U(n)$$

$U(n)$ depends upon $H(n) = \dfrac{f(n)}{n^{\log_b a}} = \dfrac{n\log n}{n^0}$

$$= n\log_2 n$$

Here n^r, $r>0$, case 1 applies.

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

Example-7

$$T(n) = 4T(n/2) + n$$

$a=4,\ b=2,\ f(n) = n$

$$T(n) = n^{\log_2 4} \cdot U(n) = n^2 \cdot U(n)$$

$U(n)$ depends upon $H(n)$

$$H(n) = \frac{f(n)}{n^{\log_b a}} = \frac{n}{n^2} = \frac{1}{n} = n^{-1}$$

$r=-1$, $r<0$, $H(n) = O(1)$

$$T(n) = n^2 \cdot O(1) = O(n^2)$$

Example-8

$$T(n) = 4T(n/2) + n^2$$

$a = 4,\ b = 2,\ f(n) = n^2$

$$T(n) = n^{\log_2 4} \cdot U(n)$$
$$= n^2 \cdot U(n)$$

$U(n)$ depends upon $H(n)$

$$H(n) = \frac{f(n)}{n^{\log_b a}} = \frac{n^2}{n^2} = 1 = (\log_2 n)^0$$

if $i \geq 0$

$$\text{then } U(n) = \frac{(\log_2 n)^{0+1}}{0+1} = O(\log_2 n)$$

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

Example-9

$$T(n) = 4T(n/2) + n^3$$
$$T(n) = n^{\log_2 4} \cdot U(n)$$
$$= n^2 \cdot U(n)$$

$U(n)$ depends upon $H(n)$

$$H(n) = \frac{f(n)}{n^{\log_b a}} = \frac{n^3}{n^2} = n^1,\ r > 0$$

$$\text{then } U(n) = O(n^1)$$

$$T(n) = n^2 \cdot O(n) = \underline{n^3}$$

Example $T(n) = 4T(n/2) + n^2 \log n$