Complexity Analysis of Recursive Algorithms: Substitution Method

Session No.: 9
Course Name: Design and analysis of algorithm
Course Code: R1UC407B
Instructor Name: Mili Dhar
Duration: 50 Min.
Date of Conduction of Class:

Review of the key concepts

1. Review of Master theorem

Q: What are the limitations of Substitution method?

Learning Outcome

Solve recurrence relations using the **Recursive tree method**.

Session Outline

1 Complexity Analysis of Recursive Algorithms using recursive tree method

2 Reflection learning activity

3 Conclusion and post-session activity

Steps to Solve Recurrence Relations Using Recursion Tree Method-

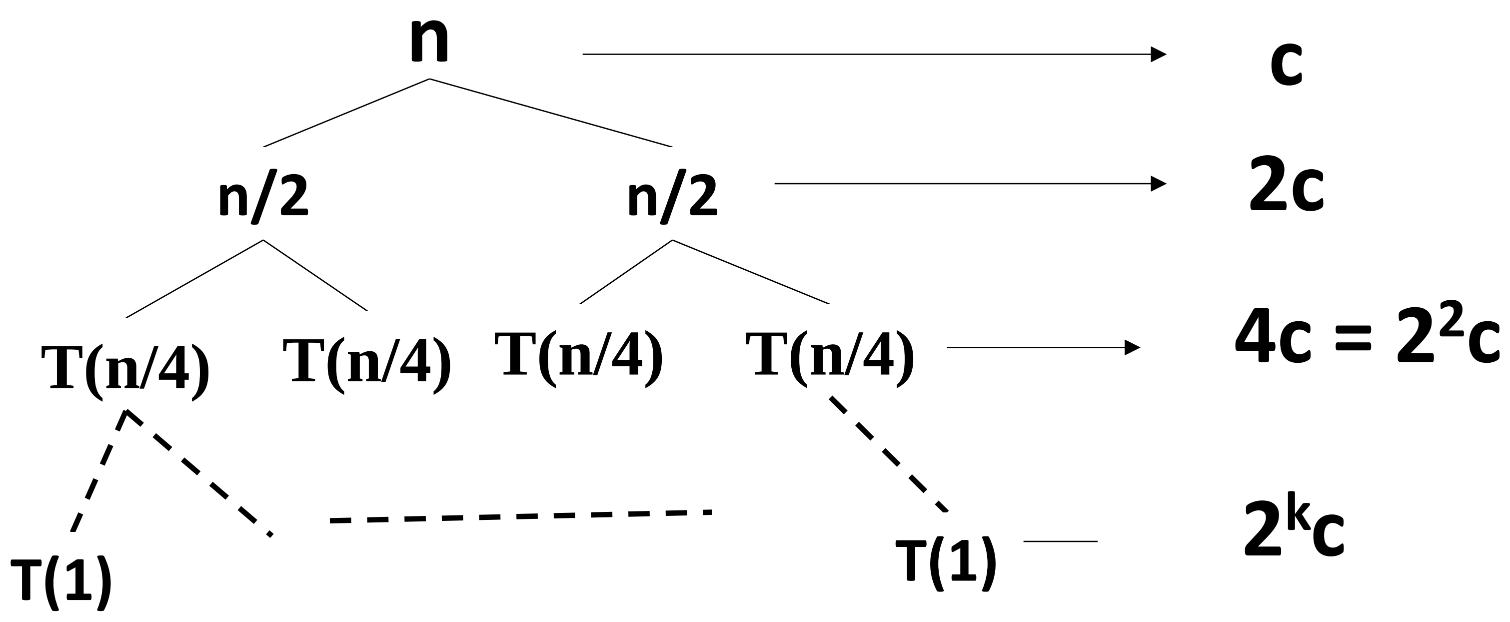
Step-01: Draw a recursion tree based on the given recurrence relation.
Step-02: Determine-

- Cost of each level
- Total number of levels in the recursion tree
- Number of nodes in the last level
- Cost of the last level

Step-03: Add the cost of all the levels of the recursion tree and simplify the expression so obtained in terms of asymptotic notation.

Recursive Tree method:

$$T(n) = 2T(n/2) + c \qquad \text{if } n>1$$
$$\quad = c \qquad \text{if } n=1$$



Assume, $n = 2^k$

Recursive Tree method:

Total cost

$c + 2c + 4c + 8c + \ldots\ldots\ldots\ldots + 2kc$

$= c\ [\ 1 + 2 + 4 + \ldots\ldots\ldots + n\ ]$

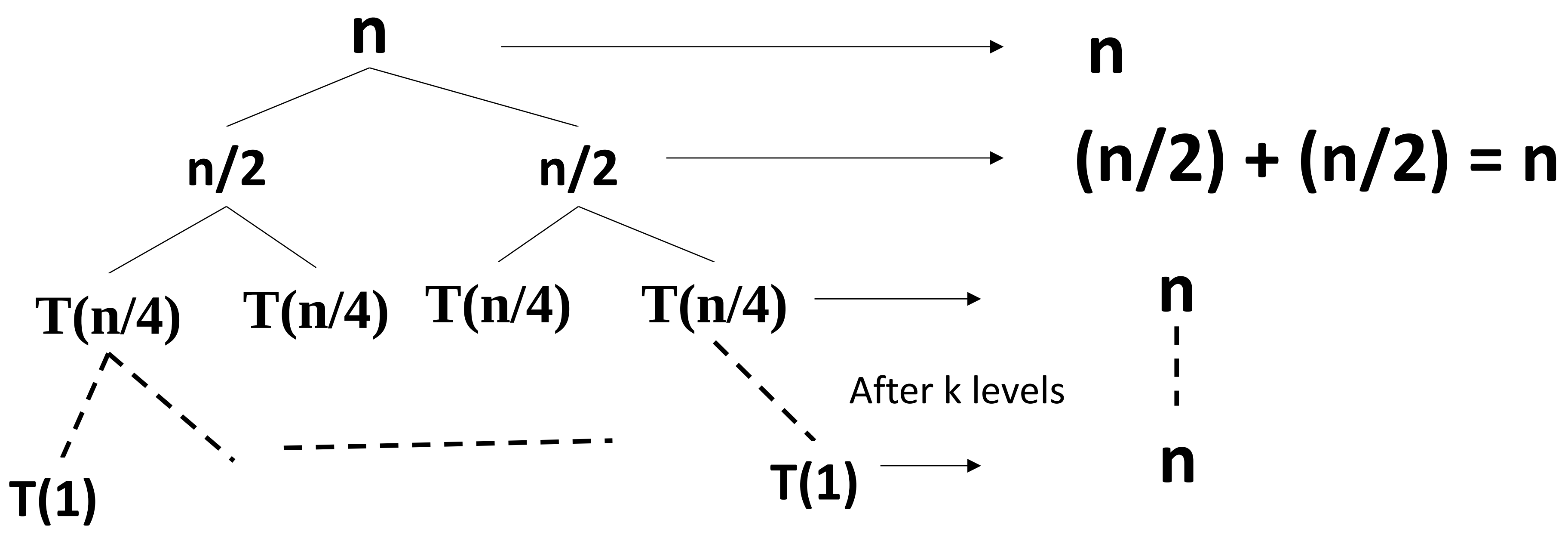
$= c\ [\ 1(n - 1)/\ (2-1)]$

$= c[n-1]$

$= cn - c$

Time complexity = $O(n)$

$$T(n) = 2T(n/2) + n \qquad \text{if } n>1$$
$$\quad\;\; = 1 \qquad \text{if } n=1$$



Assume, $n = 2^k$
➔ $k = \log n$

Total cost

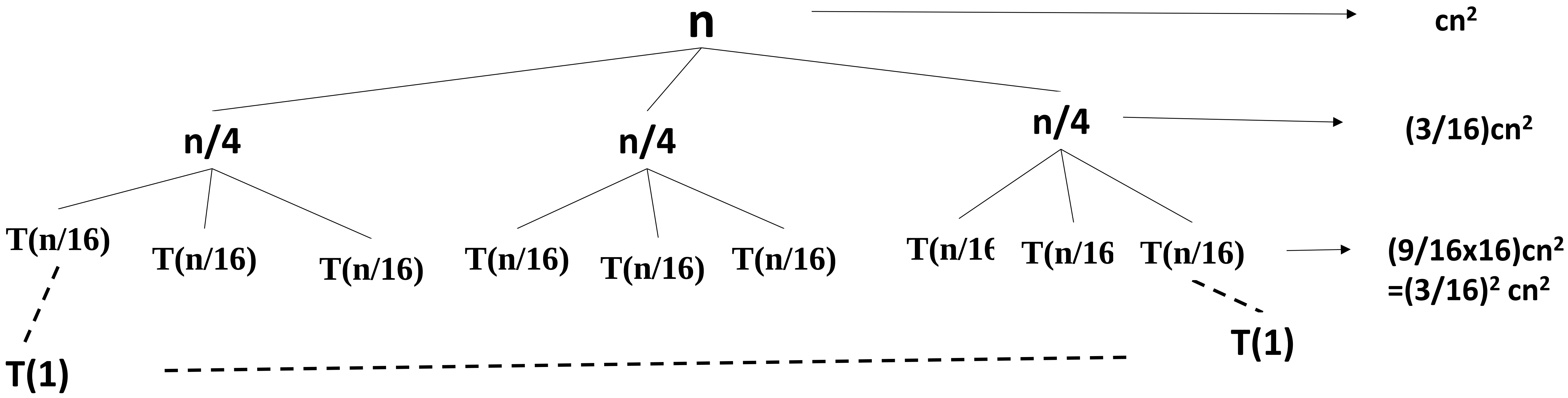
n+ n + n + n+…… k times

=kn

= n (logn)

Time complexity = O(nlogn)

$$T(n) = 3T(n/4) + cn^2 \qquad \text{if } n>1$$
$$= 1 \qquad \text{if } n=1$$



Total cost

$cn^2 + (3/16)cn^2 + (3/16)^2cn^2 + (3/16)^3cn^2 + \ldots\ldots\ldots\ldots\ldots$

$=cn^2\ [1+ (3/16) + (3/16)^2 + (3/16)^3 +\ldots\ldots\ldots]$

$=cn^2\ [1/(1-(3/16))]$

$= cn^2 \text{ x } (16/13)$

Time complexity = $O(n^2)$

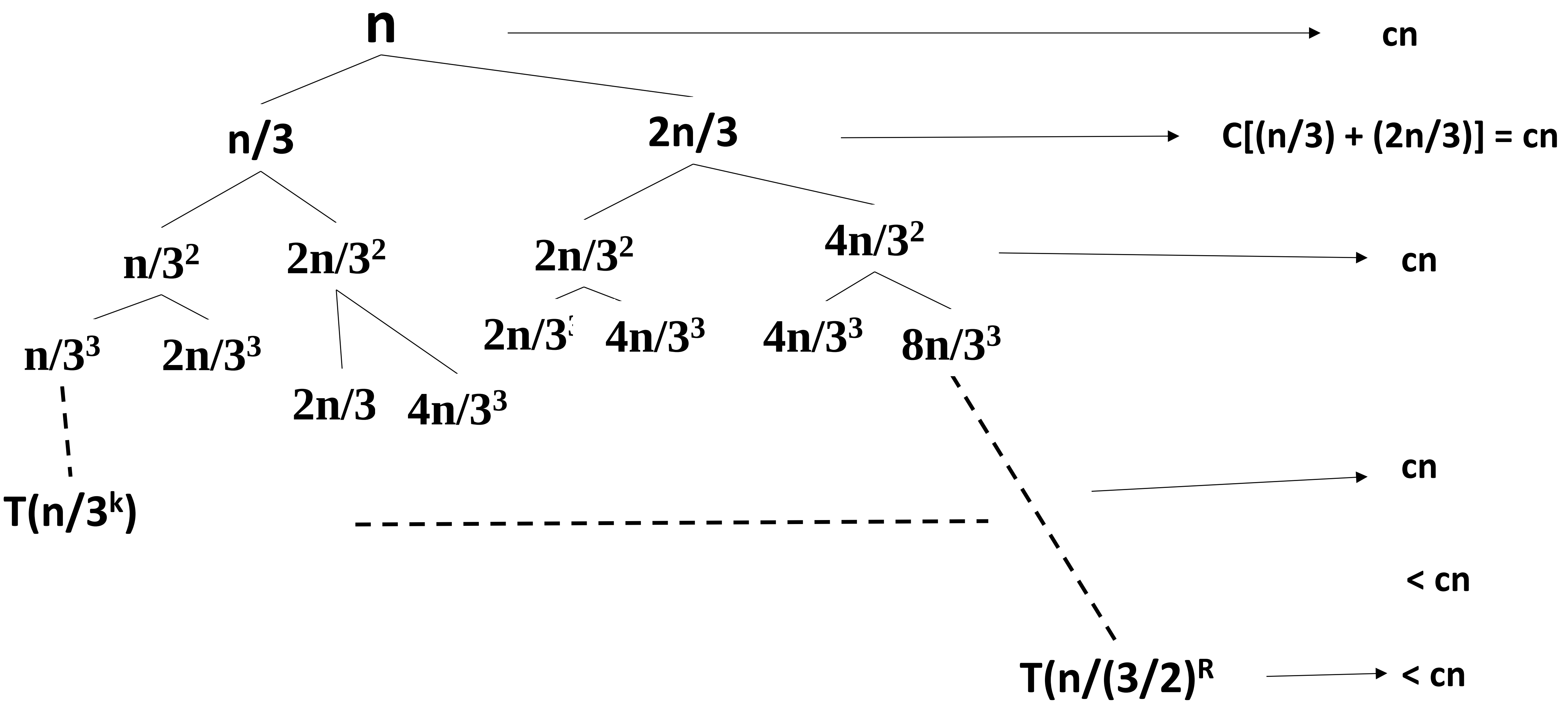
G.P for infinite series

1 + r+ r2 + r3 + …….

= 1/(1-r) if r<1

GALGOTIAS UNIVERSITY

$$T(n) = T(n/3) + T(2n/3) + cn \qquad \text{if } n>1$$
$$= 1 \qquad \text{if } n=1$$



Total cost

$=R(cn)$

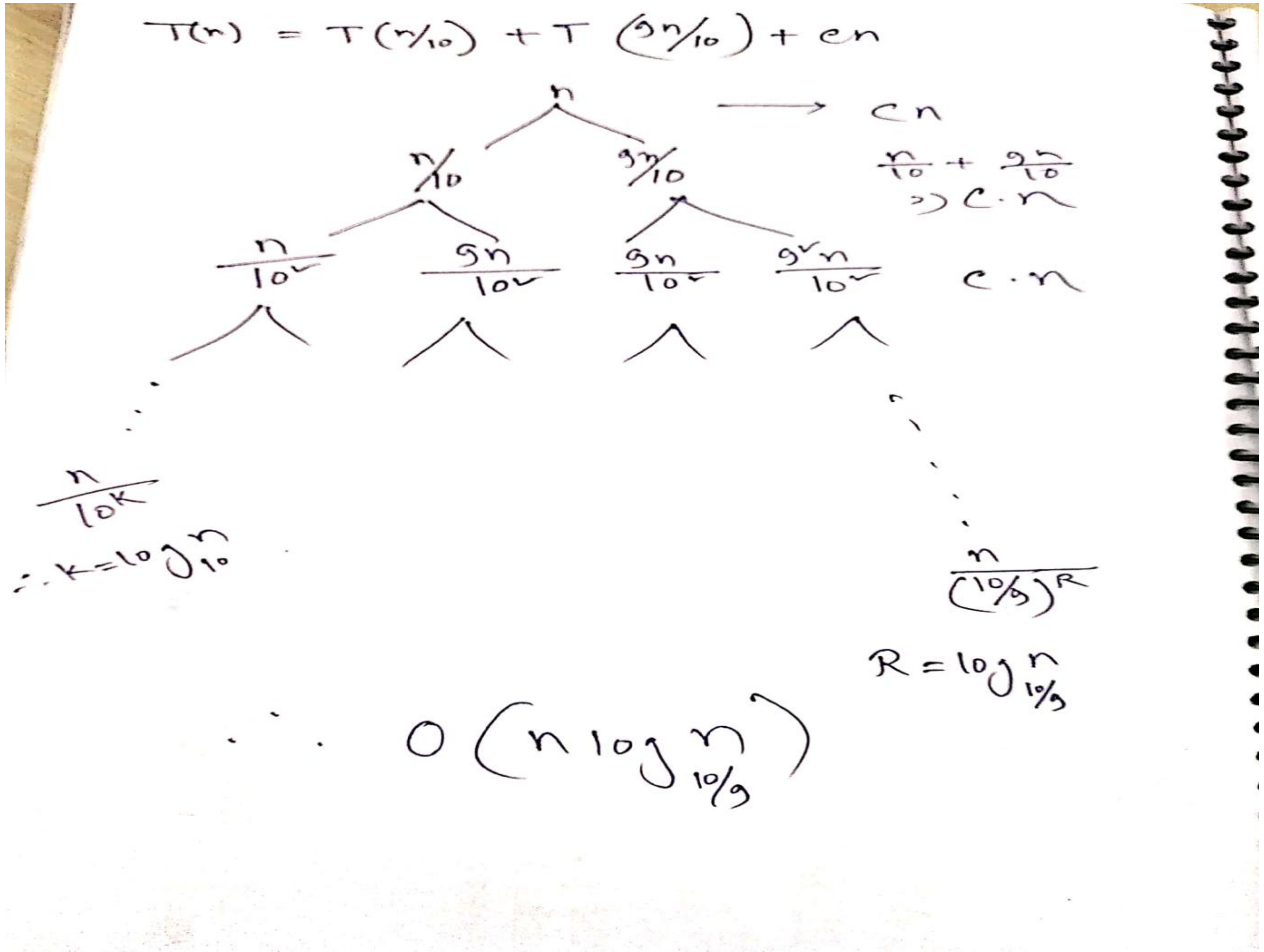
$= cn \log_{3/2} n$

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

Assume, $n= (3/2)^R$

➔ $R = \log_{3/2} n$

$$T(n) = T(n/10) + T(9n/10) + cn \qquad \text{if } n>1$$
$$= 1 \qquad \text{if } n=1$$



Post class assessment:

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

In the next class we will going through devide and conquer algorithmic approach

Thank You