Analysis and Design of Algorithms

UNIT-1

Recurrence Relations

Content

- Solving Recurrence Relations
  - **Recursion/ Recurrence Tree Method**

Recursion/ Recurrence Tree

Recurrence Tree

- We can visualize iteration method as a recursion tree in which at each level nodes are expanded.
- RECURSION TREE – "Drawing a picture of the back substitution process (iteration method) gives you a idea of what is going on".
- The recursion tree method is good for generating guesses for the substitution method.
- A recursion tree models the costs (time) of a recursive execution of an algorithm.
- The recursion-tree method promotes intuition, however.

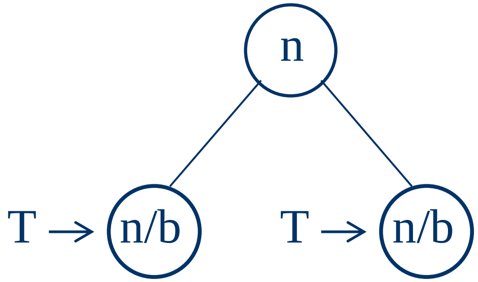
Recurrence Tree

Here while solving recurrences, we divide the problem into sub-problems of equal size.

For e.g., $\mathbf{T(n) = a\,T(n/b) + f(n)}$

where $a \geq 1$, $b > 1$ and f(n) is a given function.

F(n) is the cost of splitting or combining the sub-problems.



Recurrence Tree

- In a recursion tree ,
- each node represents the cost of a single sub-problem somewhere in the set of recursive problems invocations.
- we sum the cost within each level of the tree to obtain a set of per level cost and
- then we sum all the per level cost to determine the total cost of all levels of recursion.

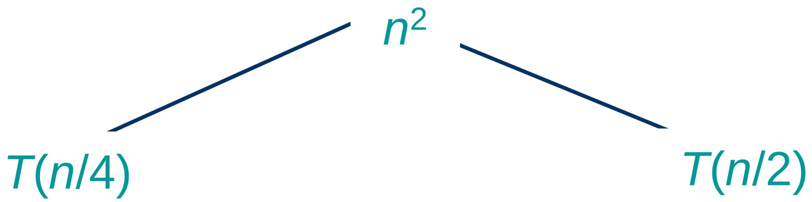
Example-1

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

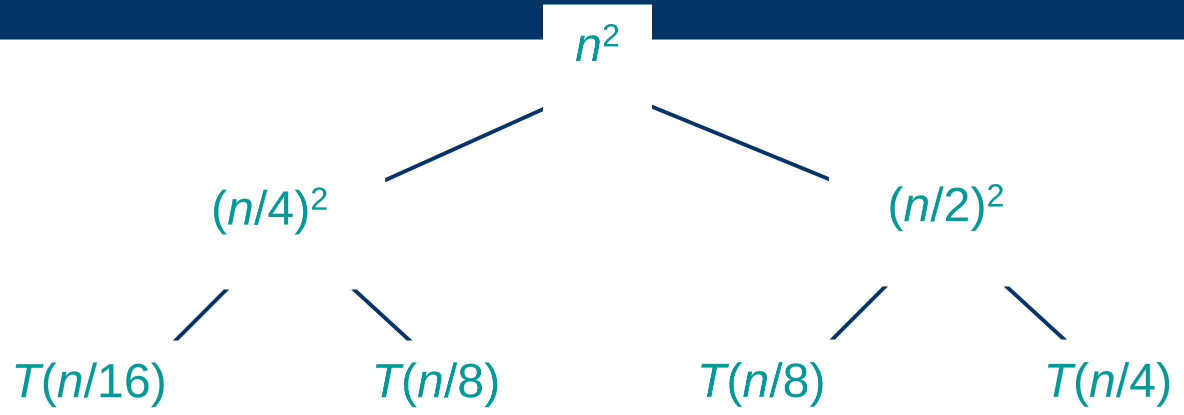
Example-1

$T(n)$

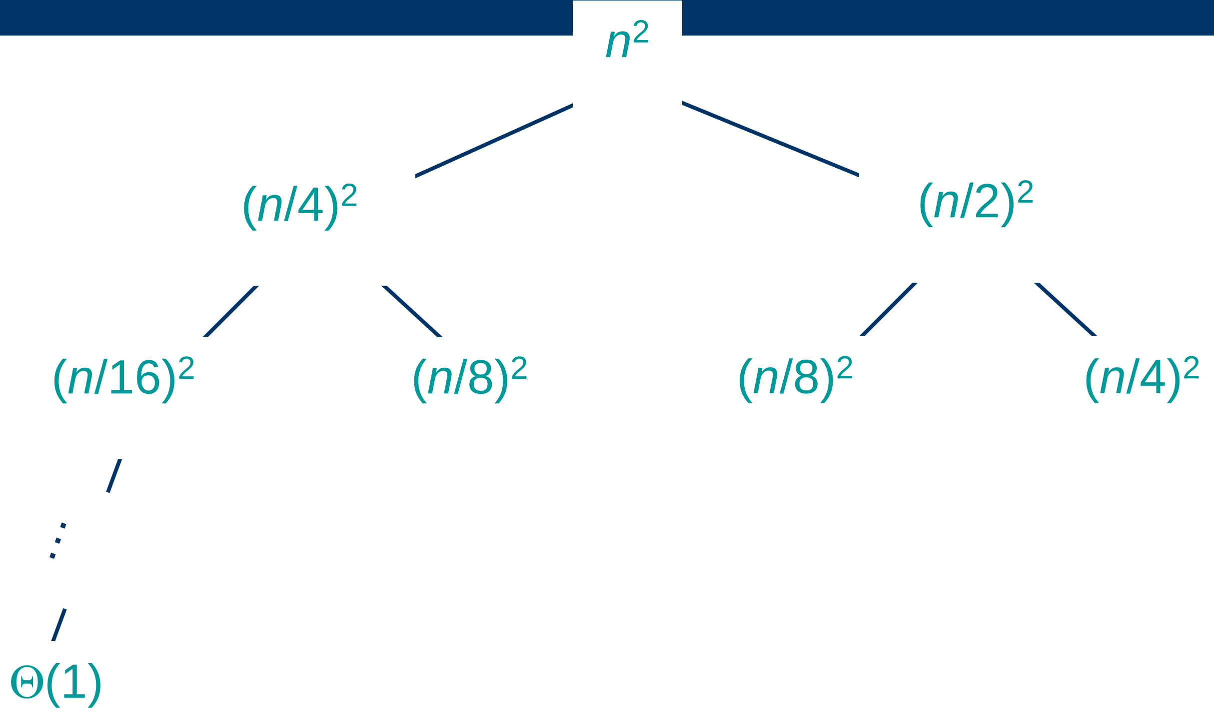
Example-1



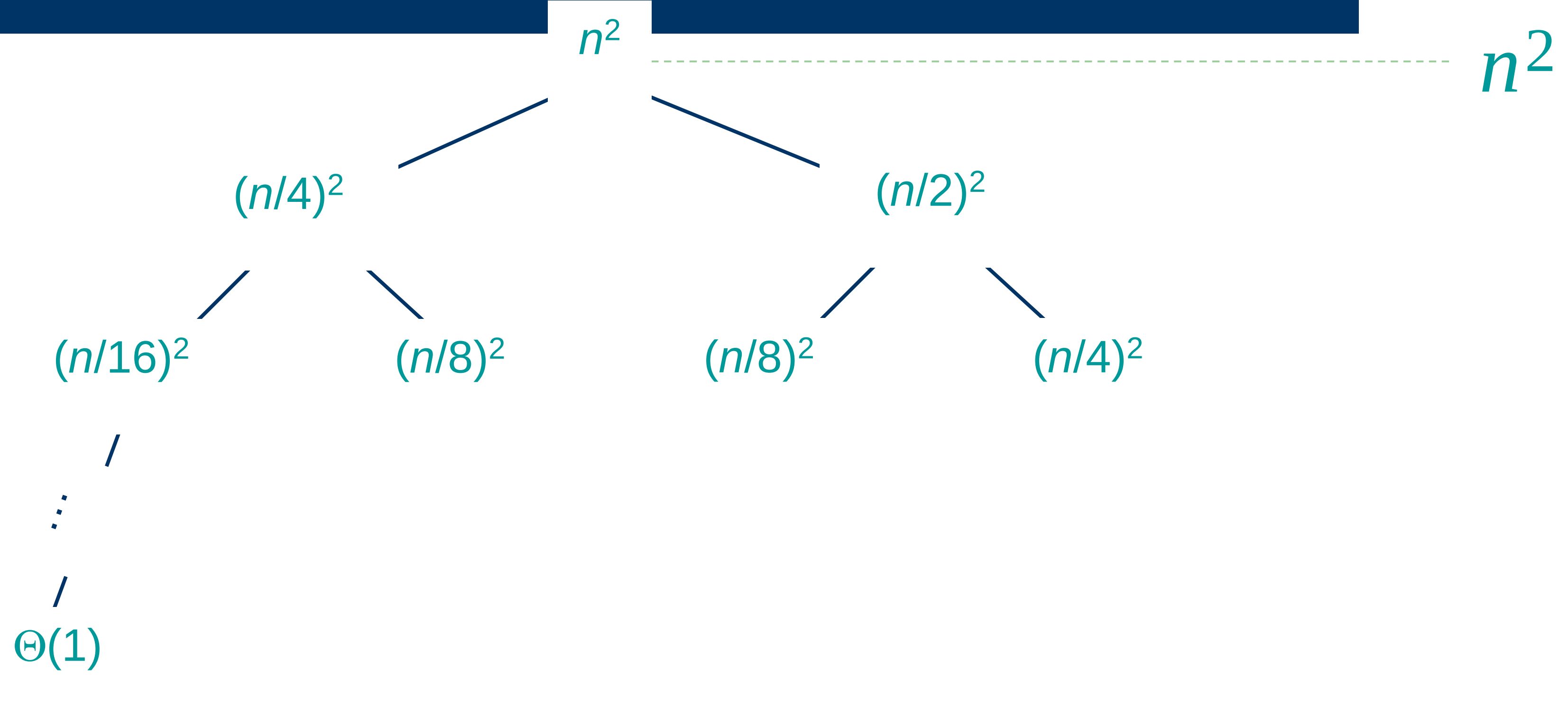
Example-1



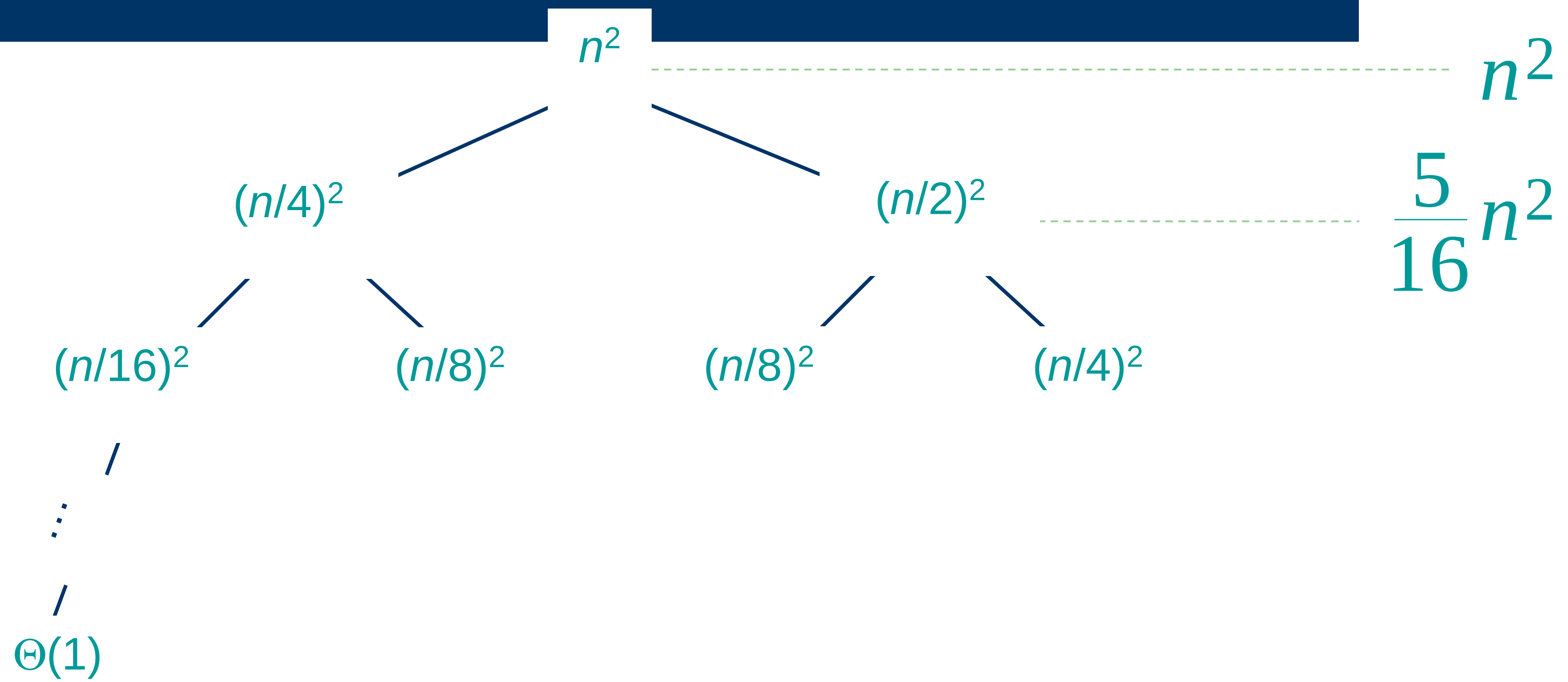
Example-1



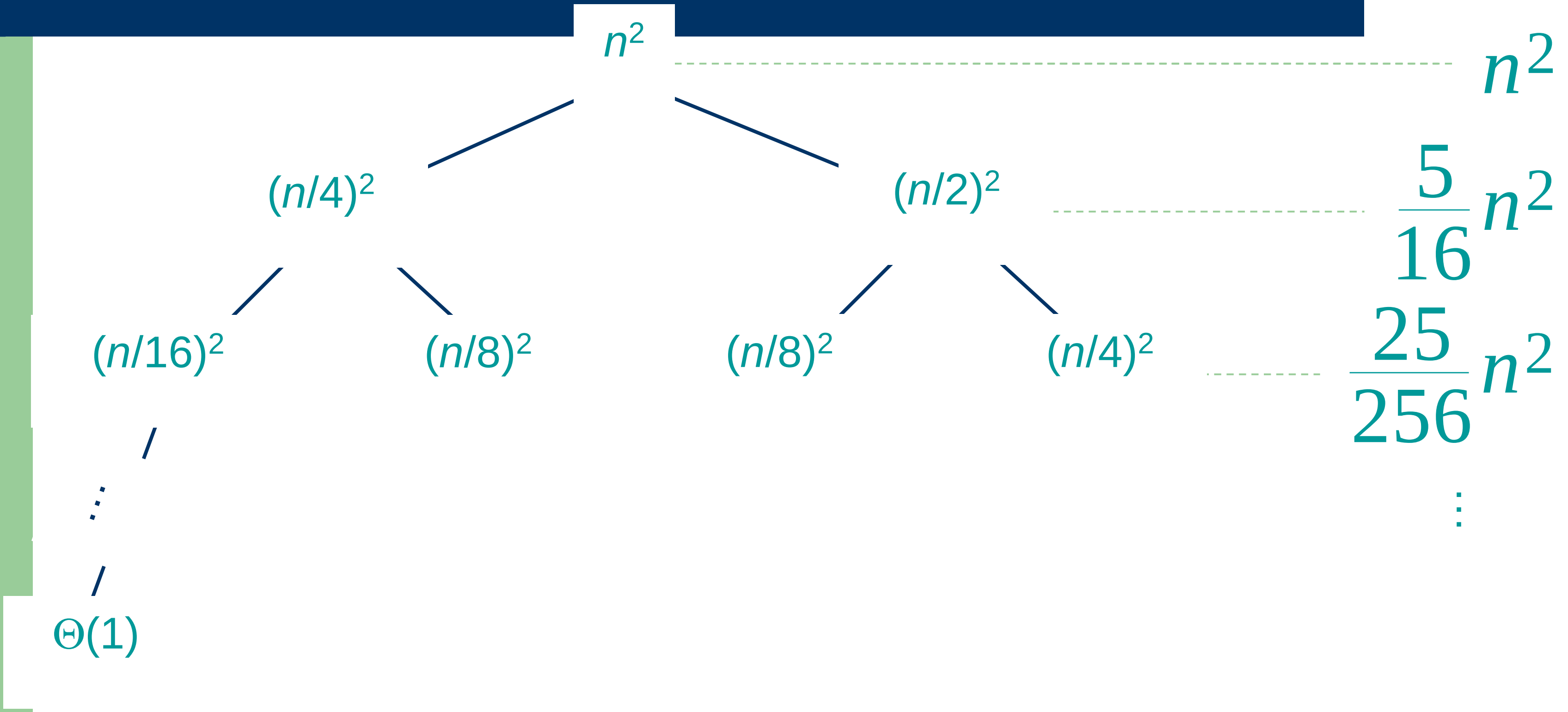
Example-1



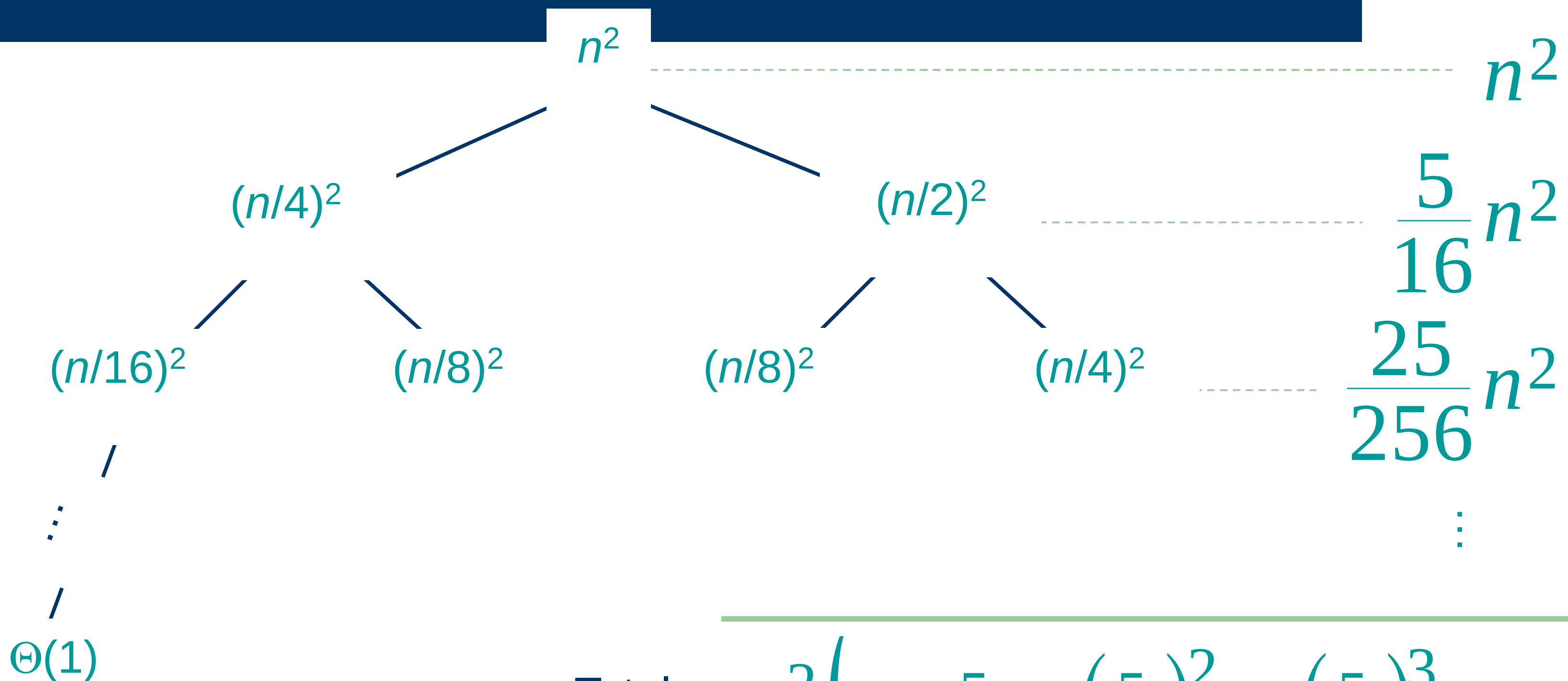
Example-1



Example-1



Example-1



Total $= n^2\left(1+\frac{5}{16}+\left(\frac{5}{16}\right)^2+\left(\frac{5}{16}\right)^3+\cdots\right)$

$= \Theta(n^2)$ *geometric series*

Example-2

- Recursion-Tree = Diagrammatic Way of Doing Iterative Expansion

Solve $T(n) = 3T(n/4) + cn^2$

Solution : on next page Diagrammatically.

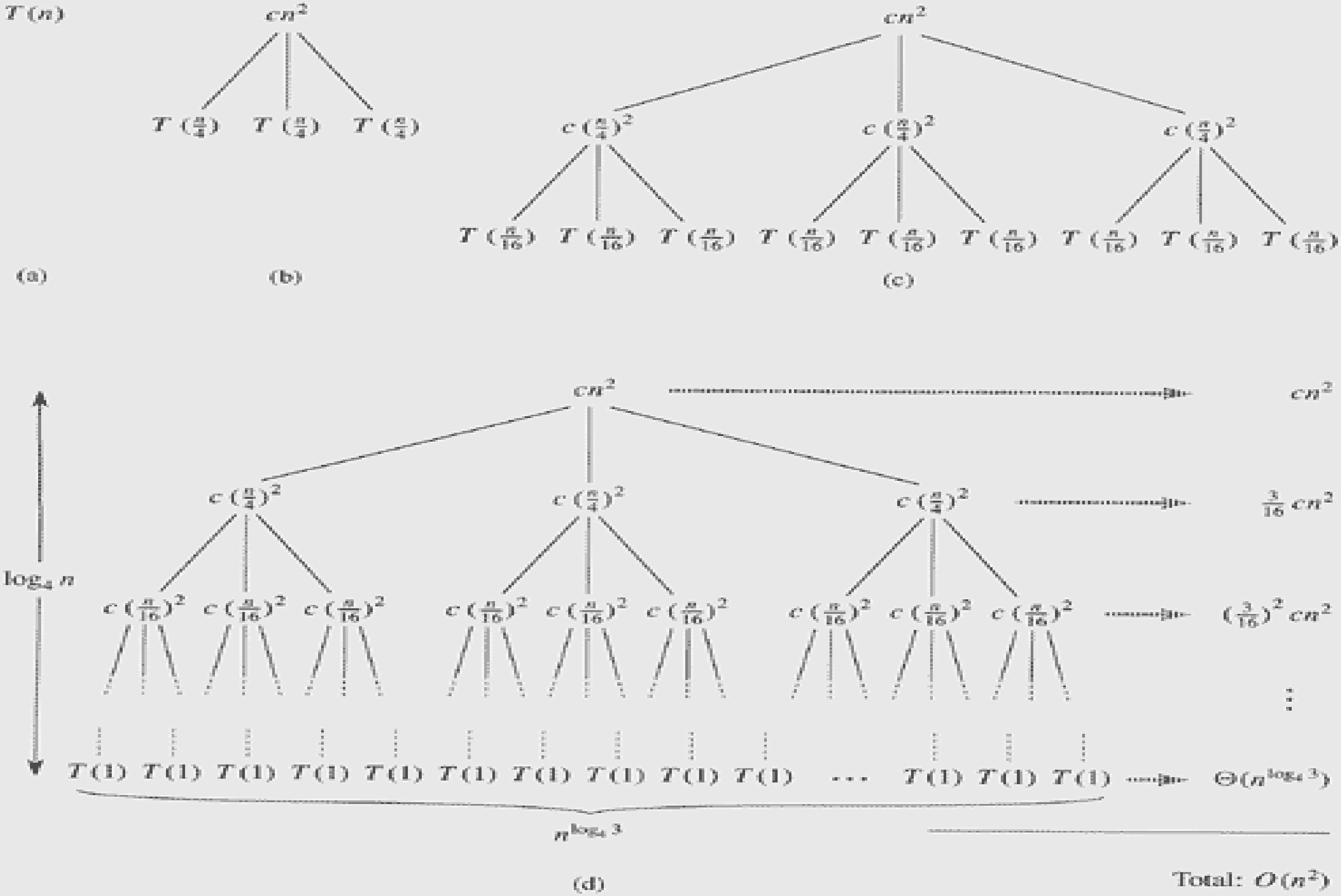


Figure 4.1 The construction of a recursion tree for the recurrence $T(n) = 3T(n/4) + cn^2$. Part **(a)** shows $T(n)$, which is progressively expanded in **(b)–(d)** to form the recursion tree. The fully expanded tree in part (d) has height $\log_4 n$ (it has $\log_4 n + 1$ levels).

Example-2

- Constructing a recursion tree for the recurrence-

  $T(n) = 3T(n/4) + c\,n^2$

- Part (a) shows T (n), which progressively expands in (b)–(d) to form the recursion tree.
- The fully expanded tree in part (d) has height-

  $\log_4 n$ (it has $\log_4 n + 1$ levels).

- Sub problem size at depth i $= n/4^i$
- Sub problem size is 1 when $n/4^i = 1 \Rightarrow i = \log_4 n$
- So, no. of levels $= 1 + \log_4 n$
- Cost of each level = (no. of nodes) x (cost of each node)

No. Of nodes at depth $i=3^i$

Cost of each node at depth $i=c\ (n/4^i)^2$

Cost of each level at depth $i=3^i c\ (n/4^i)^2 = (3/16)^i cn^2$

$T(n)= \sum_{i=0}^{\log_4 n} cn^2(3/16)^i$

$T(n)= \sum_{i=0}^{\log_4 n -1} cn^2(3/16)^i$ + cost of last level

Cost of nodes in last level $=3^i T(1)$

- ⇨ $c3^{\log_4 n}$ (at last level $i=\log_4 n$)
- ⇨ $cn^{\log_4 3}$

$$T(n)=\sum_{i=0}^{\log_4 n -1} cn^2(3/16)^i + c\,n^{\log_4 3}$$

$$<= cn^2\sum_{i=0}^{\infty} (3/16)^i + cn^{\log_4 3}$$

⇨ $<= cn^2*(16/13)+ cn^{\log_4 3} => T(n)=O(n^2)$

Example-3

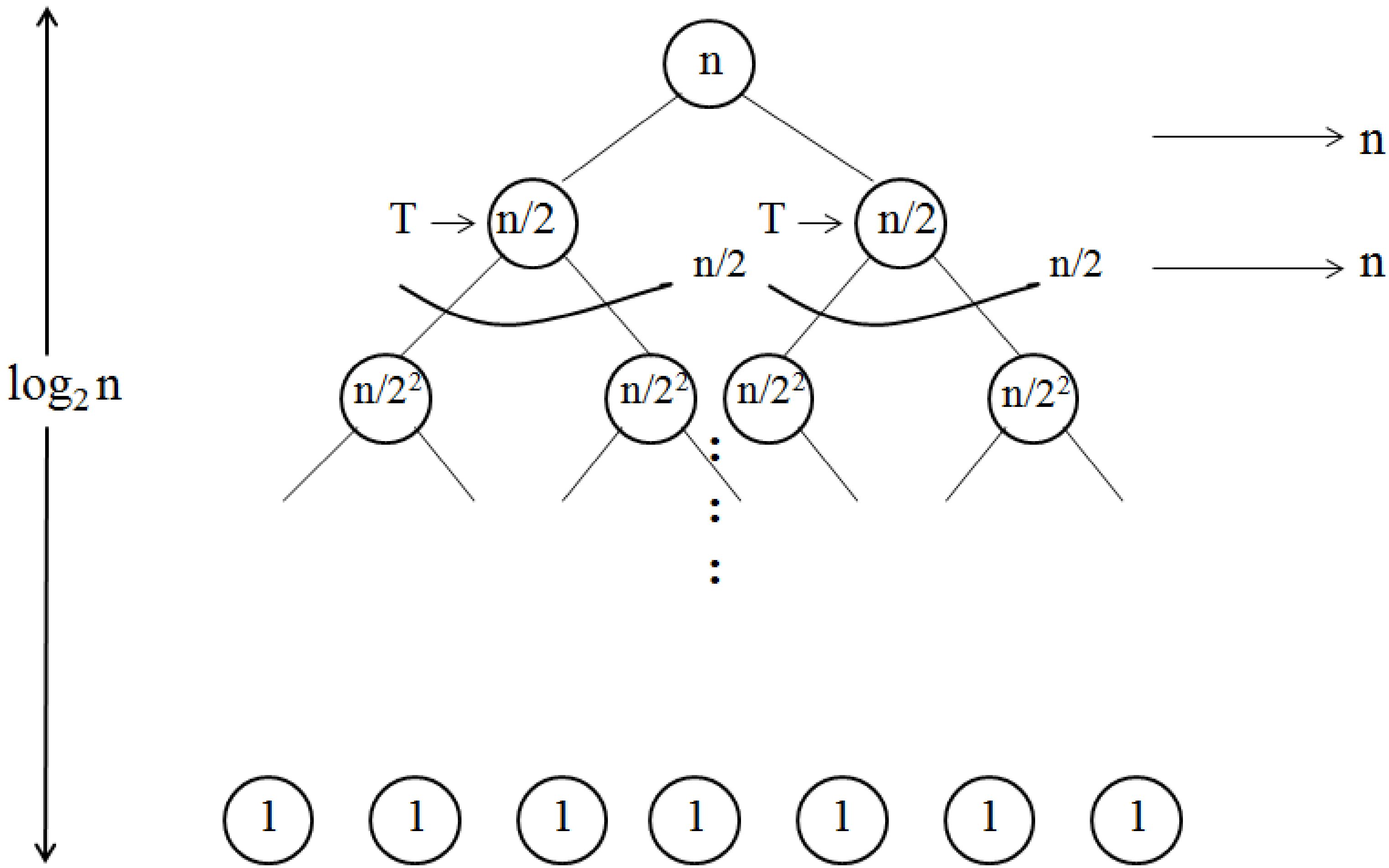
- Recursion-Tree = Diagrammatic Way of Doing Iterative Expansion

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

Solution : on next page Diagrammatically.

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

The recursion tree for this recurrence is :



Example-3

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

We have - n + n + n + …… log n times

$= n\ (1 + 1 + 1 + \ldots\ldots \text{ log n times})$

$= n\ (\log_2 n)$

$= \Theta\ (n \log n)$

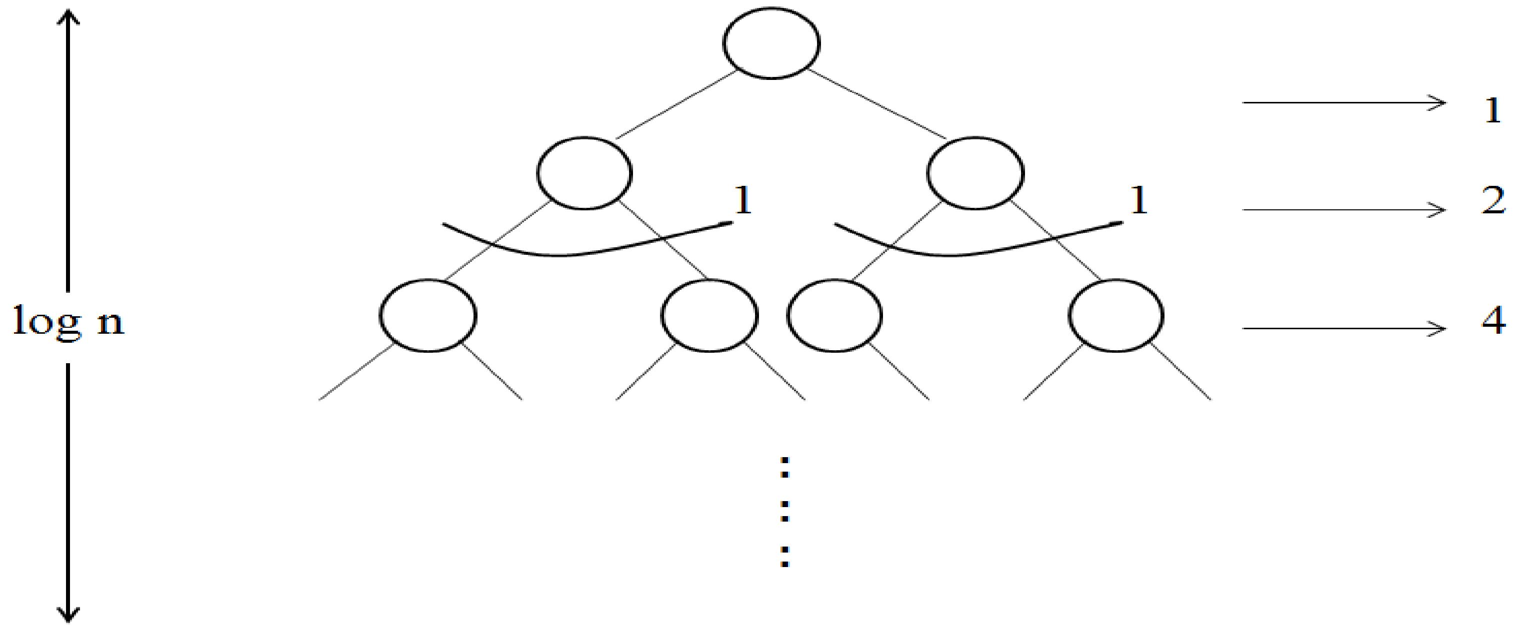
$T(n) = \Theta\ (n \log n)$

Example-3(Second Method)

II.

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

Solution : The recursion tree for the above recurrence is



Example-3(Second Method)

Now we add up the costs over all levels of the recursion tree, to determine the cost for the entire tree :

We get series like

$1 + 2 + 2^2 + 2^3 + \ldots\ldots$ log n times which is a G.P.

[So, using the formula for sum of terms in a G.P. :

$$a + ar + ar^2 + ar^3 + \ldots\ldots + ar^{n-1} = \frac{a(r^n - 1)}{r - 1} \quad]$$

$$= \frac{1\,(2^{\log n} - 1)}{2 - 1}$$

$$= n - 1$$

$$= \Theta\,(n - 1) \quad \text{(neglecting the lower order terms)}$$

$$= \Theta\,(n)$$

Example-3

Solve $T(n) = T(n/3) + T(2n/3) + cn$

Solution : on next page Diagrammatically.

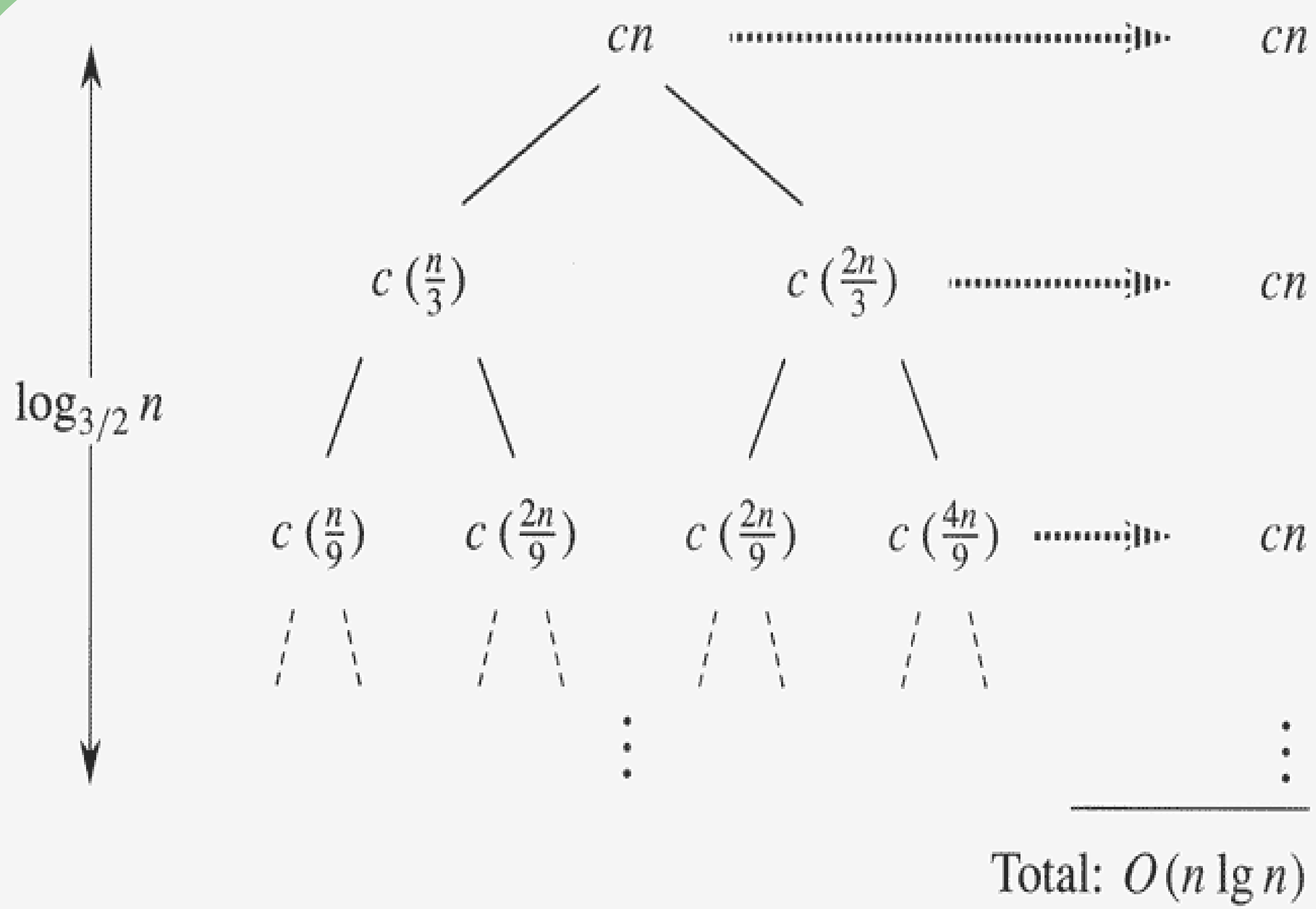


Figure 4.2 A recursion tree for the recurrence $T(n) = T(n/3) + T(2n/3) + cr$