Design and Analysis of Algorithms

Lecture - 7

Success is always inevitable with Hard Work and Perseverance

N. Ravitha Rajalakshmi

Learning Objective

- Discuss the methods of recursion tree and Master theorem
- Understand when Master theorem is applicable

Recursion Tree Method

P2

$$T(n) = 2T(n/2) + n \;, \qquad n > 1 \qquad\qquad T(1) = 1$$

Number of recursive calls

Subproblem size

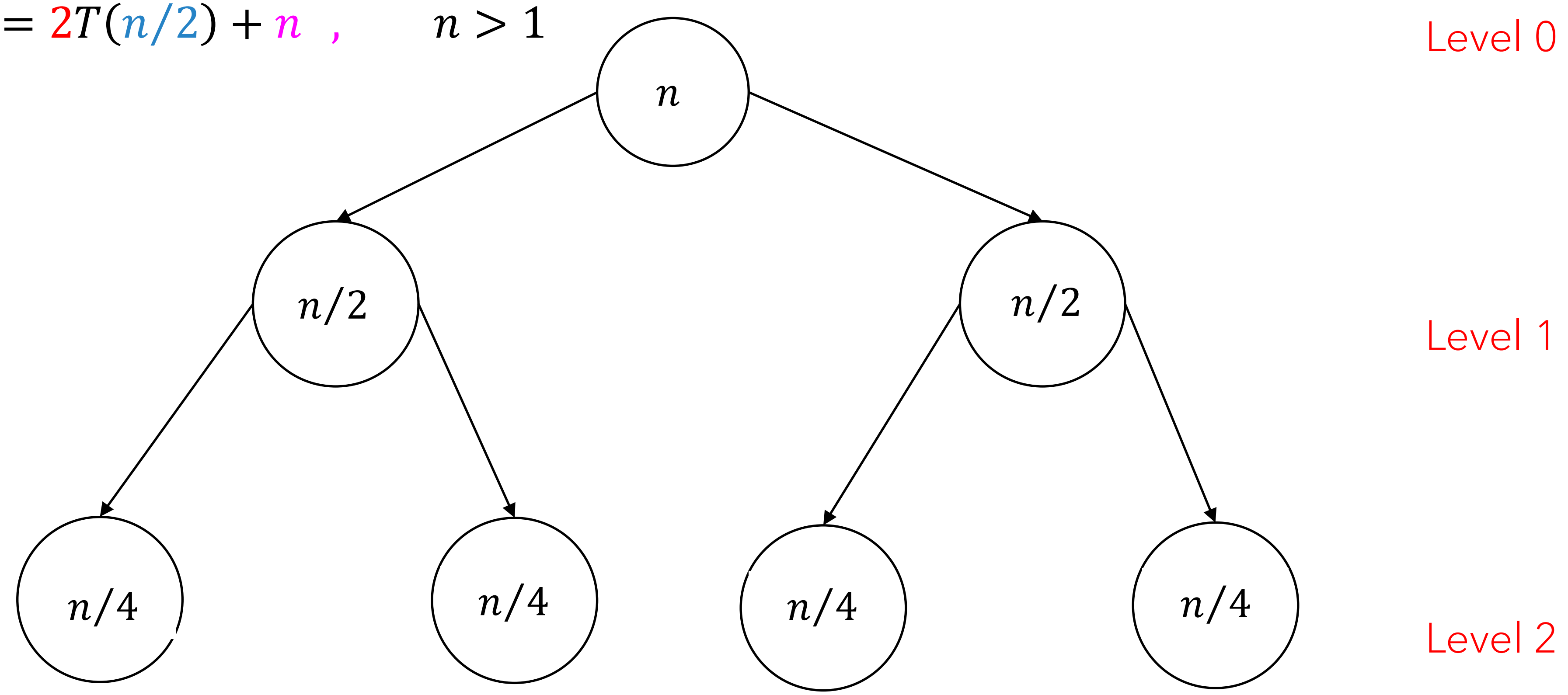
Amount of computation for input size n

Base case input size

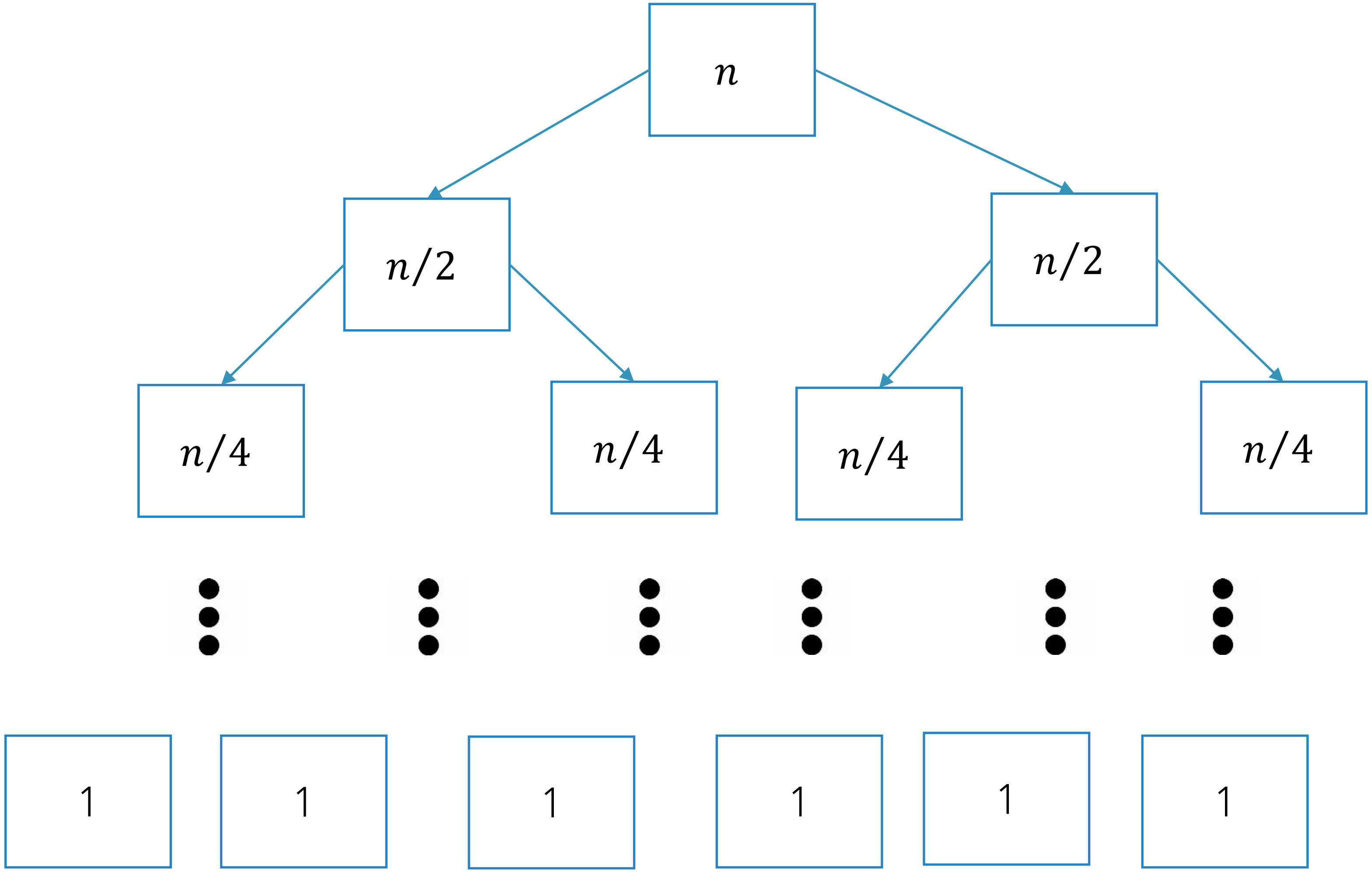
Recursion Tree Method

f(n) = n
#recursive calls = 2
Size of sub problem = n/2

$$T(n) = 2T(n/2) + n \quad , \qquad n > 1$$



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



Problem size	# nodes	Amount of work done
n	1	n
$n/2$	2	$2(n/2) = n$
$n/4$	4	$4(n/4) = n$
1	n	n * 1 = n

Cost

$$\sum_{i=a}^{b} 1 = b - a + 1$$

Cost = Cost of internal nodes + Cost of leaf nodes

= [n + n + ...] + n * 1

$$= \sum_{i=0}^{\log n - 1} n + n$$

$$= n \sum_{i=0}^{\log n - 1} 1 + n = n[\log n - 1 + 1] + n = O(n \log n)$$

Pause & Think

If problem of size (n) is reduced by a constant factor (1/3) after every recursive call, at what level does the problem get reduced to size of 1 ?

$$\frac{n}{3^{\mathrm{i}}} = 1$$

$$\mathrm{i} = \log_3 n$$

Recursion Tree Method

P3

$$T(n) = 3T(n/4) + n^2 \;, \qquad n > 1 \qquad\qquad T(1) = 1$$

Number of recursive calls

Subproblem size

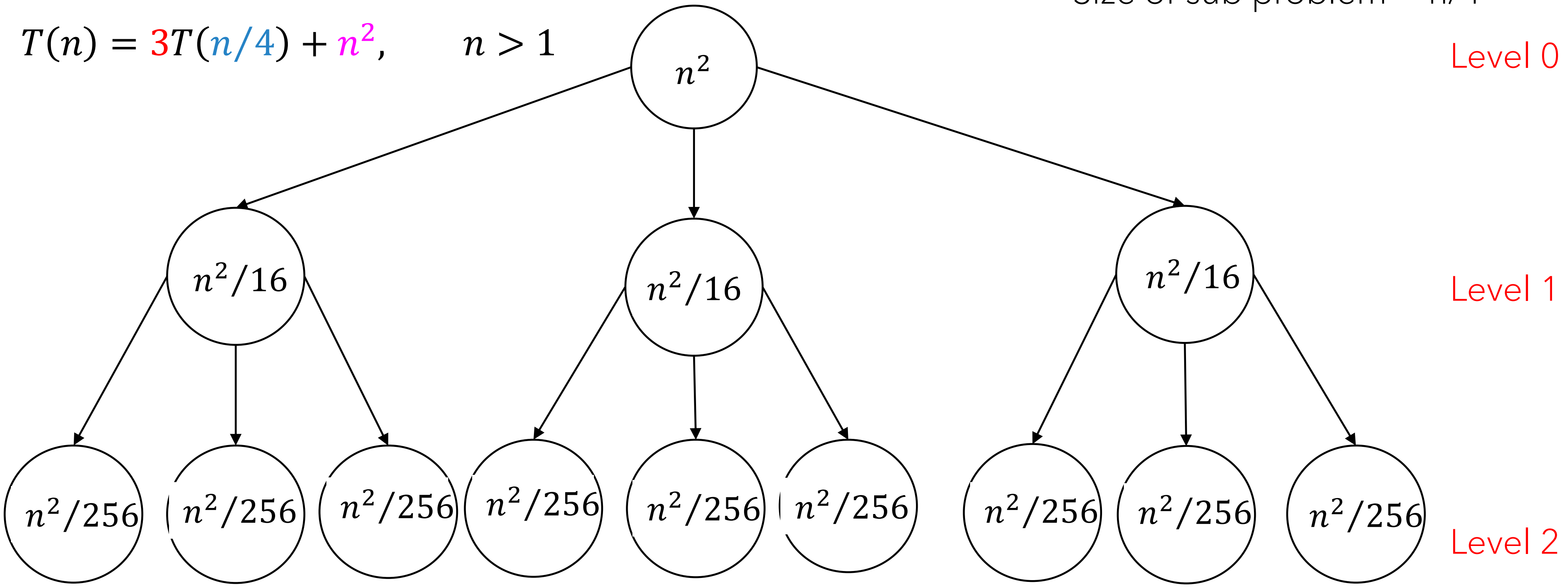
Amount of computation for input size n

Base case input size

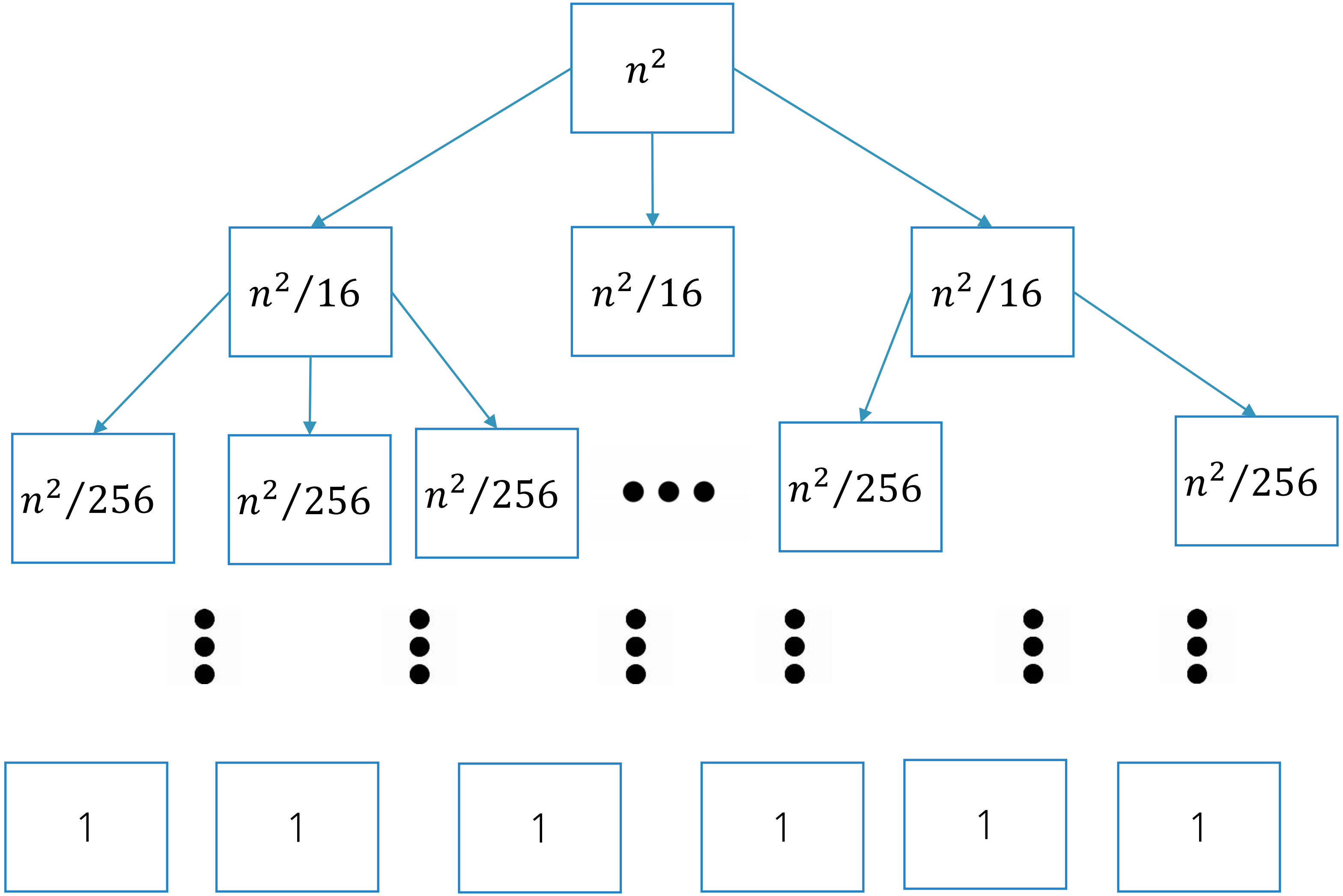
Recursion Tree Method

f(n) = n^2
#recursive calls = 3
Size of sub problem = n/4

$T(n) = 3T(n/4) + n^2, \quad n > 1$



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



Problem size	# nodes	Amount of work done
n	1	n^2
$n/4$	3	$3(n^2/16)$
$n/16$	9	$9(n^2/256)$
1	$3^{\log_4 n}$	$n^{\log_4 3}$

Cost

$\sum_{i=0}^{\infty} a^i = \frac{a}{1-a} \qquad a < 1$

Cost = Cost of internal nodes + Cost of leaf nodes

$$= [\, n^2 + 3(n^2/16) + 9(n^2/256) \ldots] + n^{\log_4 3}$$

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

$$< n^2 \sum_{i=0}^{\infty} (3/16)^i + n^{\log_4 3} = (3/13)n^2 + n^{\log_4 3} = O(n^2)$$

Is there a shortcut ?

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

$$O(n \log n)$$

Master theorem

Theorem

If $T(n) = aT(n/b) + (n^d)$ [*for constants* $a > 0$, $b > 1$, $d \geq 0$] *then*

$$T(n) = \begin{cases} O\left(n^{\log_b a}\right) & n^{\log_b a} > n^d \\ O\left(n^d \log n\right) & n^{\log_b a} = n^d \\ O\left(n^d\right) & n^{\log_b a} < n^d \end{cases}$$

a.f(n/b) < = c.f(n) for some c [0,1]

Case 3

n^d

$n^{\log_b a}$

$$T(n) = O\left(n^d\right)$$

Master theorem

P1

$$T(1) = 1$$

$$T(n) = 3T(n/4) + n^2 \quad , \qquad n > 1$$

Number of recursive calls (a)

Subproblem size (n/b)

Amount of computation for input size n (n^d)

Base case input size

a	3
b	4
d	2
n^d	n^2
$n^{\log_b a}$	$n^{\log_4 3} = n^{0.79}$

It belongs to case 3

Regularity Condition

$$a \cdot f(n/b) \leq c \cdot f(n)$$

a	3
b	4
d	2
n^d	n^2
$n^{\log_b a}$	$n^{\log_4 3} = n^{0.79}$

$$3\left(\frac{n}{4}\right)^2 \leq c \cdot n^2$$

$$\frac{3}{16}n^2 = c \cdot n^2 \qquad C = \frac{3}{16} \text{ [0.1875]}$$

It belongs to case 3 T(n) = $0(n^2)$

P2

$$T(n) = 3T(n/2) + n \quad , \qquad n > 1$$

a	3
b	2
d	1
n^d	n
$n^{\log_b a}$	$n^{\log_2 3} = n^{1.58}$

It belongs to case 1 T(n) = $O\left(n^{\log_2 3}\right)$

P3

$$T(n) = 2T(n/2) + n \quad , \qquad n > 1$$

a	2
b	2
d	1
n^d	n
$n^{\log_b a}$	$n^{\log_2 2} = n$

It belongs to case 2 T(n) = $O(n \log n)$

When Master theorem is not applicable?

If f(n) and $n^{\log_b a}$ does not differ by polynomial

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

Pause & Think

How will you prove master theorem ?

Using recursion tree

Cost = cost of root node + cost of internal node + cost of leaf node

$$n^d \quad + \quad \sum_{i=0}^{\log n} n^d \left({}^{a}/_{b^i} \right)^d + \quad n^{\log_b a}$$

Case 1: $n^{\log_b a} > n^d$ [constant multiple of n^d] (a > b^d)

Case 2: $n^d = n^{\log_b a}$ $[n^d * \sum_{i=0}^{\log n} (1)^i] = n^d \log n$

Case 3: $n^d > n^{\log_b a}$ [a.f(n/b) <= c.f(n)] (a < b^d)

Summary

- For problems that are reduced by constant factor on every recursive call with polynomial computation, learnt a shortcut for assessing time complexity.

Thank You
Happy Learning

Success is always inevitable with Hard Work and Perseverance