Design and Analysis of Algorithms

Lecture - 8

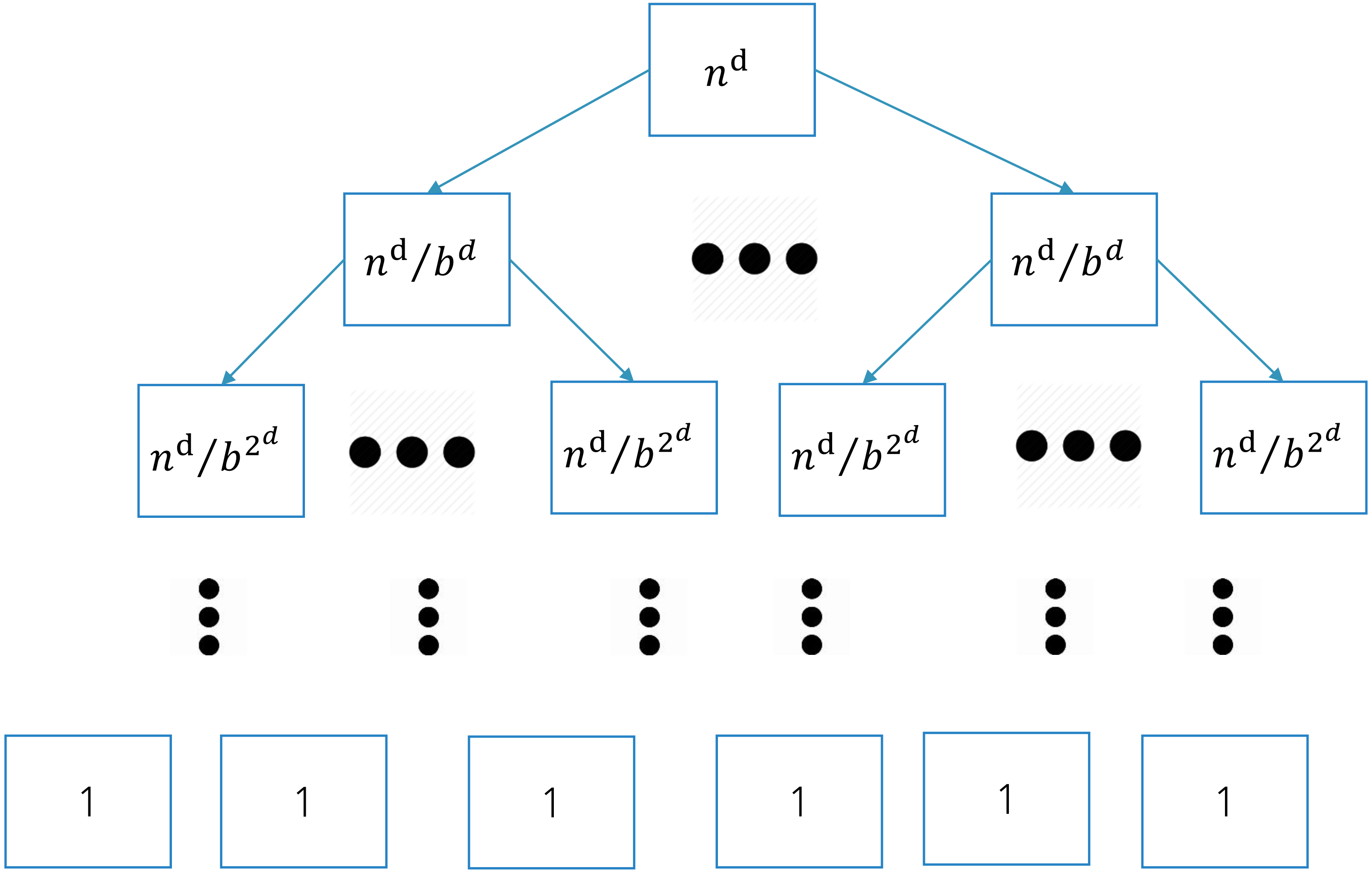
Success is always inevitable with Hard Work and Perseverance

N. Ravitha Rajalakshmi

Learning Objective

- Learn the proof of Master theorem
- Discuss the recurrences when Master theorem is not applicable

$T(n) = aT(b) + n^d$



Problem size	# nodes	Amount of work done
n	1	n^d
n/b	a	$a\, n^d/b^d$
n/b^2	a^2	$a^2\, n^d/b^{2^d}$
n/b^i	a^i	$a^i\, n^d/b^{i^d}$
l	$n^{\log_b a}$	$n^{\log_b a}$

Pause & Think

How will you prove master theorem ?

Using recursion tree

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

$$\sum_{i=0}^{\log n-1} n^d \left({}^{a}/_{b^i}\right)^d + \quad n^{\log_b a}$$

$$\sum_{i=0}^{\log n-1} n^d \left({}^{a}/_{b^d}\right)^i + \quad n^{\log_b a}$$

$$r = \frac{a}{b^d} \quad , \qquad a = n^d$$

$$\sum_{i=0}^{n} ar^n = \begin{cases} O(a) & r < 1 \\ O(ar^{n-1}) & r > 1 \end{cases}$$

$$\sum_{i=0}^{\log n - 1} n^d \left({}^{a}/_{b^d}\right)^i + \quad n^{\log_b a}$$

Case 1 : $\log_b a > d \qquad \frac{a}{b^d} > 1$

$$n^d \left(\frac{a}{b^d}\right)^{\log n} + n^{\log_b a}$$

$$n^d \frac{a^{\log n}}{(b^d)^{\log n}} + n^{\log_b a} = n^d \frac{a^{\log n}}{(n^d)^{\log b}} + n^{\log_b a} = a^{\log n} + n^{\log_b a} = O(n^{\log_b a})$$

$$\sum_{i=0}^{\log n-1} n^d \left({}^{a}/_{b^d}\right)^i + \quad n^{\log_b a}$$

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

Case 2 : $\log_b a = d \qquad \frac{a}{b^d} = 1$

$$= \sum_{i=0}^{\log n-1} n^d (1)^i + \quad n^{\log_b a}$$

$$= n^d \sum_{i=0}^{\log n-1} (1)^i + \quad n^{\log_b a}$$

$$= n^d \log n + \quad n^{\log_b a}$$

$$= \mathrm{O}(n^d \log n)$$

$$\sum_{i=0}^{\log n-1} n^d \left({}^{a}/{}_{b^d}\right)^i + \quad n^{\log_b a}$$

Case 3 : $\log_b a < \ d \qquad \frac{a}{b^d} < 1$

$$\sum_{i=0}^{n} a(r^i) = \text{a} \quad \text{if r} < 1$$

$$= \sum_{i=0}^{\log n-1} n^d \left({}^{a}/{}_{b^d}\right)^i + \quad n^{\log_b a}$$

$$= n^d + \quad n^{\log_b a}$$

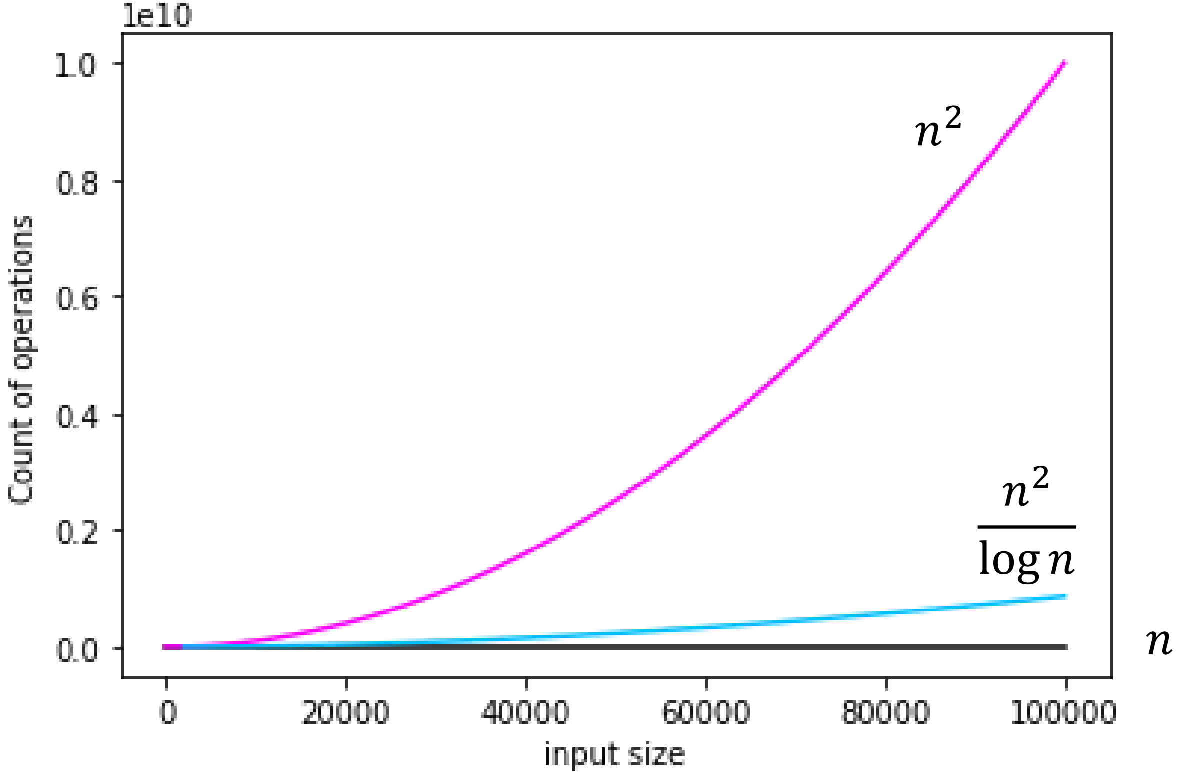
$$= \text{O}(n^d)$$

Why Regularity Condition?

$$T(n) = T(n/2) + \frac{n^2}{\log n}, \qquad n > 1$$

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

$O(n^2)$



Check for regularity condition

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

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

$$\frac{n^2}{4 * \log(n/2)} <= c \cdot \frac{n^2}{\log(n)}$$

$$\frac{n^2}{4 * \log(n/2)} > c \cdot \frac{n^2}{\log(n)}$$

$$\log(n/2) << \log(n)$$

Regularity condition not satisfied

When Master theorem is not applicable?

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

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

f(n) +ve function

a and b are non negative integers a > 0 b>1

Worksheet problem

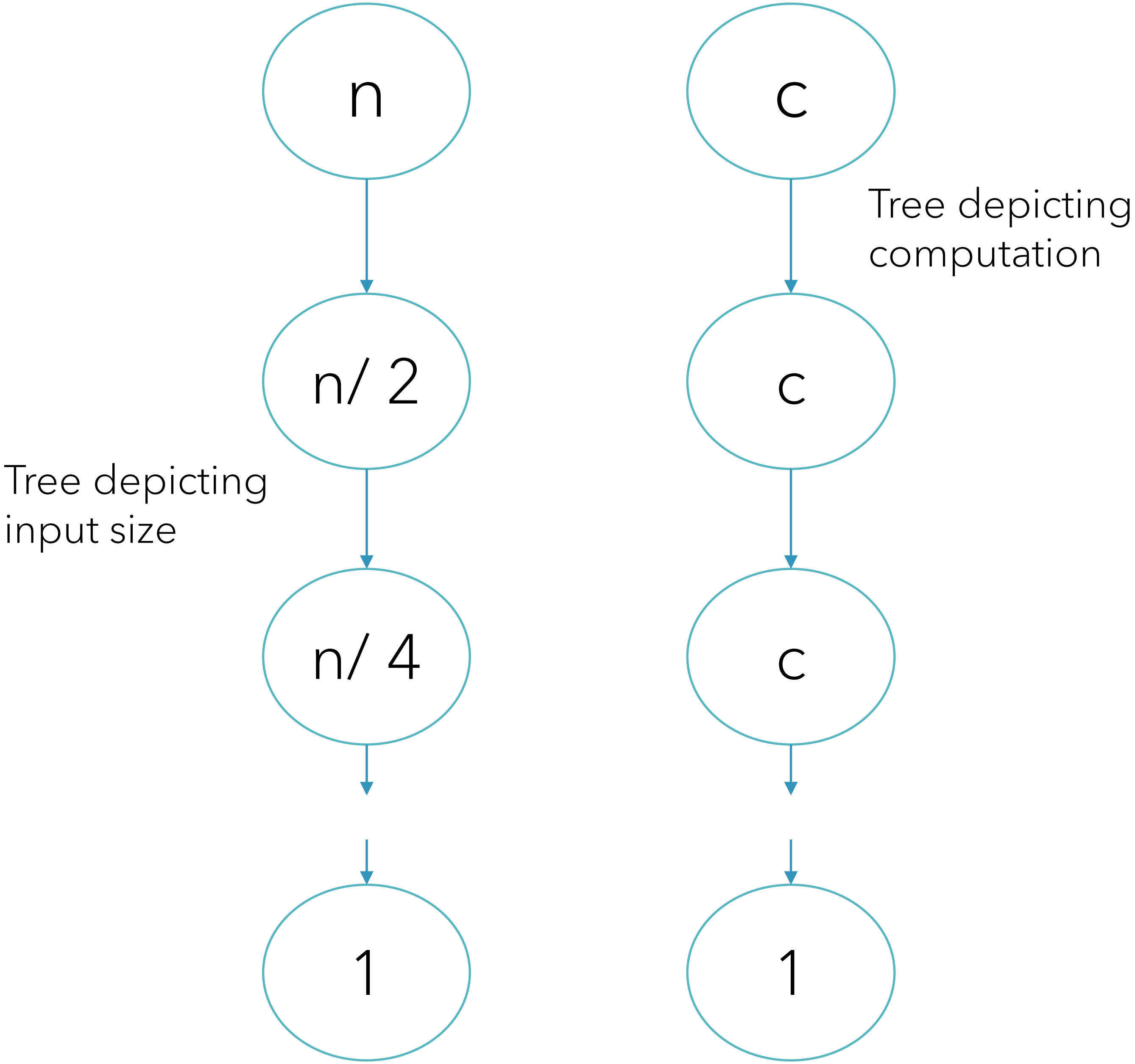
$T(n) = T(n/2) + c$ c can be any integer

C indicates fixed number of computations. It does not vary with input size

If Problem size is 2 , number of computation is c

Size = n # computations = c

Size = n/2 # computations = c



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

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

C $(\log n)$

$\mathrm{T(n)} = \mathrm{O(\log n)}$

Permutation Vs Combination

Different ways in which elements can be selected

Given a map of stations, find shortest path between a and b and do not visit the intermediate stations more than once [a……………...z]

Path acdefb , adcefb, acdfeb

Ordering matters = Permutation

Select subset of items which maximize the revenue items are {a,b,c..........f}

Subsets acd aeb abcdef

Ordering doesn't matter = Combination

Given n elements, number of subsets = 2^n

Given n elements find all possible permutations of say r elements $= np_r = \frac{n!}{(n-r)!}$

Given n elements find all possible combinations of say r elements $= nC_r = \frac{n!}{r!\,.(n-r)!}$

Fibonacci mod m

For every m , Fib mod m produces a repeating sequence which starts with {0 1 ….} of length l

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
F_i	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610
$F_i \bmod 2$	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1	0
$F_i \bmod 3$	0	1	1	2	0	2	2	1	0	1	1	2	0	2	2	1

Length of the sequence is known as piasno period

Even for very large numbers , piasno period is too small

Summary

- Discussed proof of master theorem
- Discussed on combinatorial problems and complexity of Brute force techniques

Thank You
Happy Learning

Success is always inevitable with Hard Work and Perseverance