CS 3100
Data Structures and Algorithms 2
Lecture 9: D&C, Master Theorem

Co-instructors: Robbie Hott and Tom Horton

Fall 2023

Readings in CLRS 4th edition:

- Section 4.5

Announcements

- Upcoming dates
  - PS2 due September 29 (Friday) at 11:59pm
  - PA2 due October 8 (Sunday) at 11:59pm
  - Quiz 1 and 2, October 5 (in-class)
- Course email (comes to both professors and head TAs):

cs3100@cshelpdesk.atlassian.net

Divide and Conquer

[CLRS Chapter 4]

Divide:

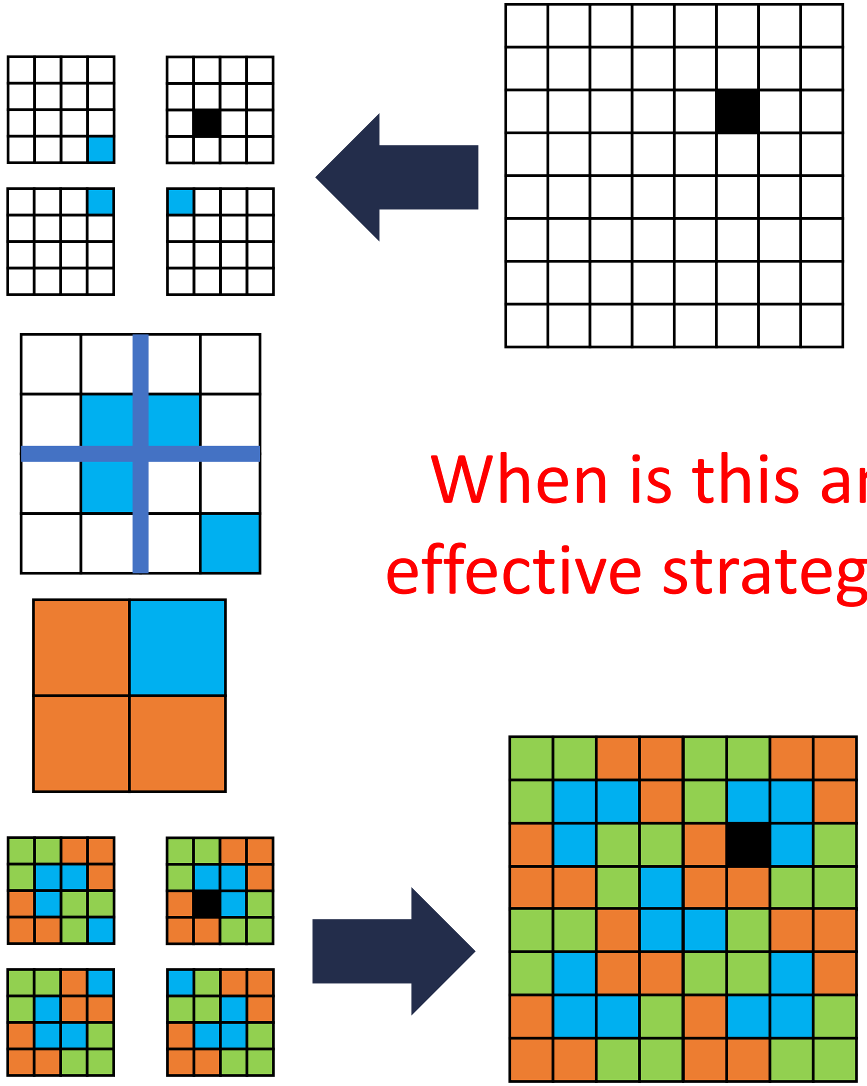
- Break the problem into multiple subproblems, each smaller instances of the original

Conquer:

- If the suproblems are "large":
  - Solve each subproblem recursively
- If the subproblems are "small":
  - Solve them directly (base case)

Combine:

- Merge solutions to subproblems to obtain solution for original problem



When is this an effective strategy?

Analyzing Divide and Conquer

1. Break into smaller subproblems
2. Use recurrence relation to express recursive running time
3. Use asymptotic notation to simplify

Divide: $D(n)$ time

Conquer: Recurse on smaller problems of size $s_1, \ldots, s_k$

Combine: $C(n)$ time

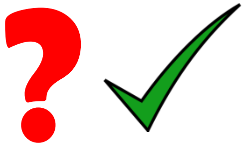
Recurrence:

- $T(n) = D(n) + \sum_{i \in [k]} T(s_i) + C(n)$

Recurrence Solving Techniques

Tree

get a picture of recursion



Guess/Check

guess and use induction to prove

“Cookbook”

MAGIC!

Substitution

substitute in to simplify

Induction (Review)

Goal: $\forall k \in \mathbb{N}, P(k)$ holds

Base case(s): $P(1)$ holds

Technically, called *strong induction*

Hypothesis: $\forall x \leq x_0, P(x)$ holds

Inductive step: $P(1), \ldots, P(x_0) \Rightarrow P(x_0 + 1)$

Guess and Check Blueprint

Show: $T(n) = O(g(n))$

Consider: $g_*(n) = c \cdot g(n)$ for some constant c

Goal: show $\exists n_0$ such that $\forall n > n_0, T(n) \leq g_*(n)$

- (definition of big-O)

Technique: Induction

- Base cases:
  - Show $T(1) \leq g_*(1)$ (sometimes, may need to consider additional base cases)
- Hypothesis:
  - $\forall n \leq x_0, T(n) \leq g_*(n)$
- Inductive step:
  - Show that $T(x_0 + 1) \leq g_*(x_0 + 1)$

Need to ensure that in inductive step, can either appeal to a base case or to the inductive hypothesis

Karatsuba Analysis using Guess and Check

$$T(n) = 3T(n/2) + 8n$$

Goal: $T(n) \leq 3000\, n^{1.6} = O(n^{1.6})$

Base case: $T(1) = 8 \leq 3000$

Hypothesis: $\forall n \leq x_0,\ \ T(n) \leq 3000n^{1.6}$

Inductive step: Show $T(x_0 + 1) \leq 3000(x_0 + 1)^{1.6}$

Karatsuba Guess and Check (Loose)

$$T(n) = 3T(n/2) + 8n$$

Hypothesis: $\forall n \leq x_0$: $T(n) \leq 3000n^{1.6}$

Show: $T(x_0 + 1) \leq 3000(x_0 + 1)^{1.6}$

$$T(x_0 + 1) = 3T\left(\frac{x_0 + 1}{2}\right) + 8(x_0 + 1)$$ Recurrence definition

$$\leq 3\left(3000\left(\frac{x_0 + 1}{2}\right)^{1.6}\right) + 8(x_0 + 1)$$ Inductive hypothesis

Karatsuba Guess and Check (Loose)

$$T(x_0+1) = 3T\left(\frac{x_0+1}{2}\right) + 8(x_0+1)$$ Recurrence definition

$$\leq 3\left(3000\left(\frac{x_0+1}{2}\right)^{1.6}\right) + 8(x_0+1)$$ Inductive hypothesis

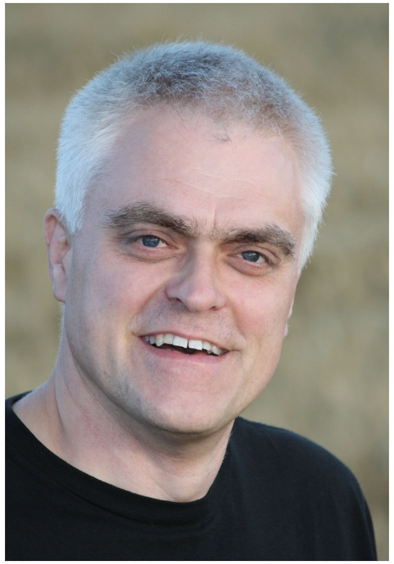
$$\leq 3\left(3000\left(\frac{x_0+1}{2}\right)^{1.6}\right) + 8(x_0+1)^{1.6}$$ $\forall x \geq 0: x^{1.6} \geq x$

$$= \left(\frac{9000}{2^{1.6}} + 8\right)(x_0+1)^{1.6}$$ Distributive property

$$\leq 3000(x_0+1)^{1.6}$$ $\frac{9000}{2^{1.6}} + 8 \leq 3000$

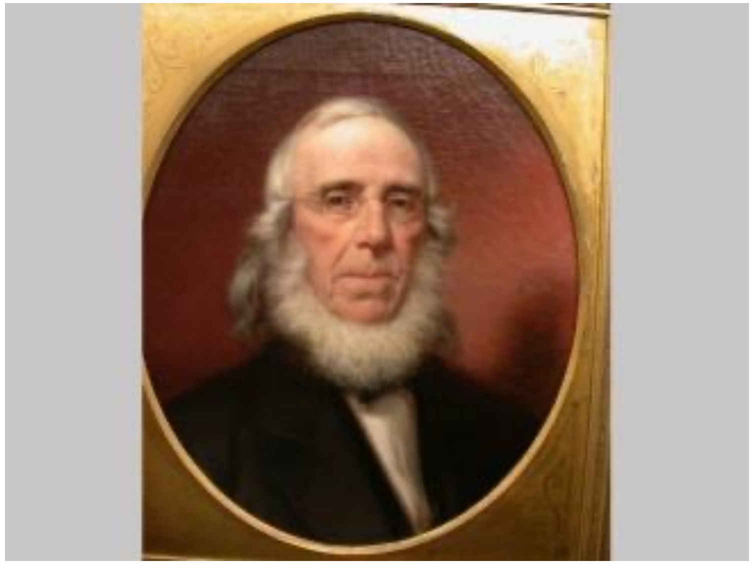
Show: $T(x_0+1) \leq 3000(x_0+1)^{1.6}$

Master Theorem for Recurrences



Jon Bentley

Dorothea Blostein
(née Haken)



James B. Saxe

Observation

Divide: $D(n)$ time

Conquer: Recurse on smaller problems of size $s_1, \ldots, s_k$

Combine: $C(n)$ time

Recurrence:

- $T(n) = D(n) + \sum_{i \in [k]} T(s_i) + C(n)$

Many divide and conquer algorithms have recurrences are of form:

- $T(n) = a \cdot T(n/b) + f(n)$

a and b are constants

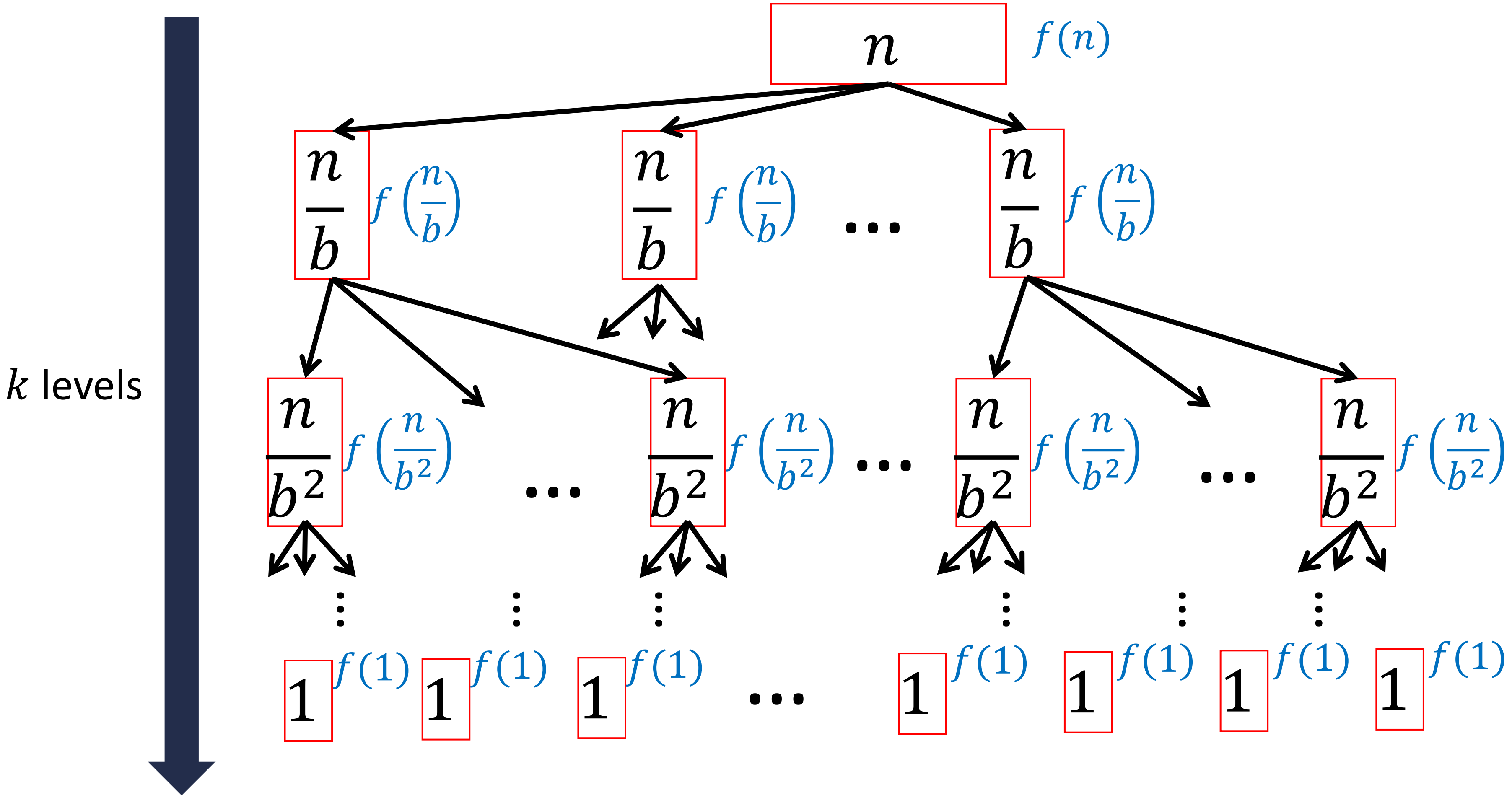
Mergesort: $T(n) = 2T(n/2) + n$

Divide and Conquer Multiplication: $T(n) = 4T(n/2) + 5n$

Karatsuba Multiplication: $T(n) = 3T(n/2) + 8n$

General Recurrence

$$T(n) = aT(n/b) + f(n)$$



	Number of subproblems	Cost of subproblem
	1	$f(n)$
	a	$f(n/b)$
	a^2	$f(n/b^2)$
	a^k	$f(n/b^k)$

General Recurrence

3. Use asymptotic notation to simplify

$$T(n) = aT(n/b) + f(n)$$

Number of subproblems	Cost of subproblem
1	$f(n)$
a	$f(n/b)$
a^2	$f(n/b^2)$
a^k	$f(n/b^k)$

How many levels?

Problem size at k^{th} level: $\frac{n}{b^k}$

Base case: $n = 1$

At level k, it should be the case that $\frac{n}{b^k} = 1$

$$n = b^k \Rightarrow k = \log_b n$$

General Recurrence

3. Use asymptotic notation to simplify

$$T(n) = aT(n/b) + f(n)$$

$$k = \log_b n$$

What is the cost?

Cost at level i: $a^i \cdot f\left(\frac{n}{b^i}\right)$

Total cost: $T(n) = \sum_{i=0}^{\log_b n} a^i \cdot f\left(\frac{n}{b^i}\right)$

Number of subproblems	Cost of subproblem
1	$f(n)$
a	$f(n/b)$
a^2	$f(n/b^2)$
a^k	$f(n/b^k)$

Three Cases

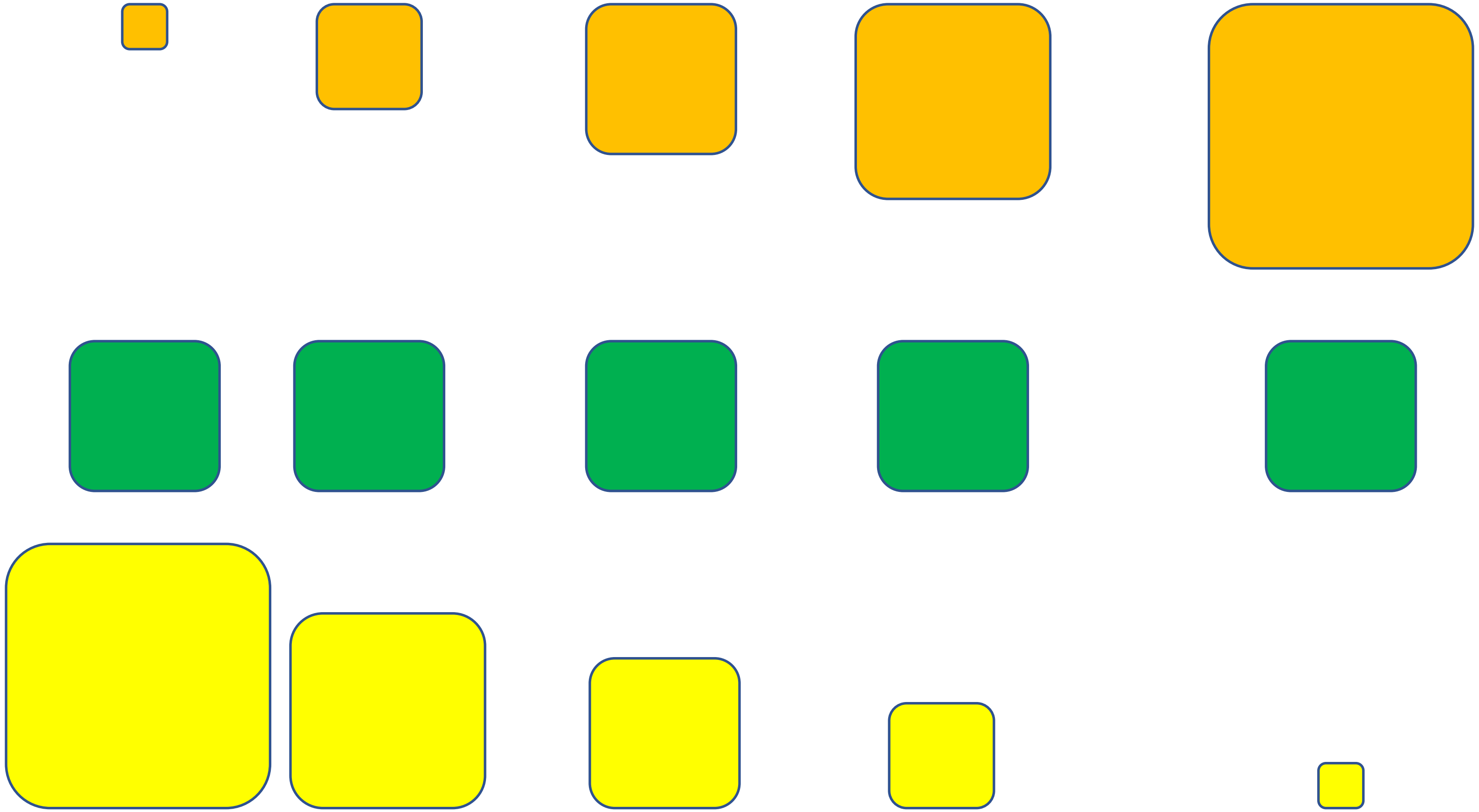
$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^k f\left(\frac{n}{b^k}\right)$$

$$k = \log_b n$$

Case 1:
Most work happens
at the leaves

Case 2:
Work happens
consistently throughout

Case 3:
Most work happens
at top of tree



Master Theorem

$$T(n) = aT(n/b) + f(n)$$

$$\delta = \log_b a$$

	Requirement on f	Implication
Case 1	$f(n) \in O\left(n^{\delta-\varepsilon}\right)$ for some constant $\varepsilon > 0$	$T(n) \in \Theta\left(n^{\delta}\right)$

Master Theorem

$$T(n) = aT(n/b) + f(n)$$

$$\delta = \log_b a$$

	Requirement on f	Implication
Case 1	$f(n) \in O\left(n^{\delta-\varepsilon}\right)$ for some constant $\varepsilon > 0$	$T(n) \in \Theta\left(n^{\delta}\right)$
Case 2	$f(n) \in \Theta\left(n^{\delta}\right)$	$T(n) \in \Theta\left(n^{\delta} \log n\right)$

Master Theorem

$$T(n) = aT(n/b) + f(n)$$

$$\delta = \log_b a$$

	Requirement on f	Implication
Case 1	$f(n) \in O\left(n^{\delta-\varepsilon}\right)$ for some constant $\varepsilon > 0$	$T(n) \in \Theta\left(n^{\delta}\right)$
Case 2	$f(n) \in \Theta\left(n^{\delta}\right)$	$T(n) \in \Theta\left(n^{\delta} \log n\right)$
Case 3	$f(n) \in \Omega\left(n^{\delta+\varepsilon}\right)$ for some constant $\varepsilon > 0$ **AND** $af\left(\frac{n}{b}\right) \leq cf(n)$ for constant $c < 1$ and sufficiently large n	$T(n) \in \Theta\left(f(n)\right)$

Master Theorem Example 1

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

[Merge Sort]

$$T(n) = aT(n/b) + f(n)$$

$$\delta = \log_b a$$

Master Theorem Example 1

$$T(n) = 2T(n/2) + n$$ [Merge Sort]

Step 1: Compute $\delta = \log_b a = \log_2 2 = 1$

Step 2: Compare n^δ and $f(n)$

$$f(n) = n \in \Theta(n^\delta)$$

Step 3: Check table

$T(n) = aT(n/b) + f(n)$ $\delta = \log_b a$

Master Theorem Example 1

$\delta = 1$

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

[Merge Sort]

$f(n) = n \in \Theta(n^{\delta})$

	Requirement on f	Implication
Case 1	$f(n) \in O(n^{\delta-\varepsilon})$ for some constant $\varepsilon > 0$	$T(n) \in \Theta(n^{\delta})$
Case 2	$f(n) \in \Theta(n^{\delta})$	$T(n) \in \Theta(n^{\delta} \log n)$
Case 3	$f(n) \in \Omega(n^{\delta+\varepsilon})$ for some constant $\varepsilon > 0$ **AND** $af\left(\frac{n}{b}\right) \leq cf(n)$ for constant $c < 1$ and sufficiently large n	$T(n) \in \Theta(f(n))$

Master Theorem Example 1

$$T(n) = 2T(n/2) + n$$ [Merge Sort]

$\delta = 1$

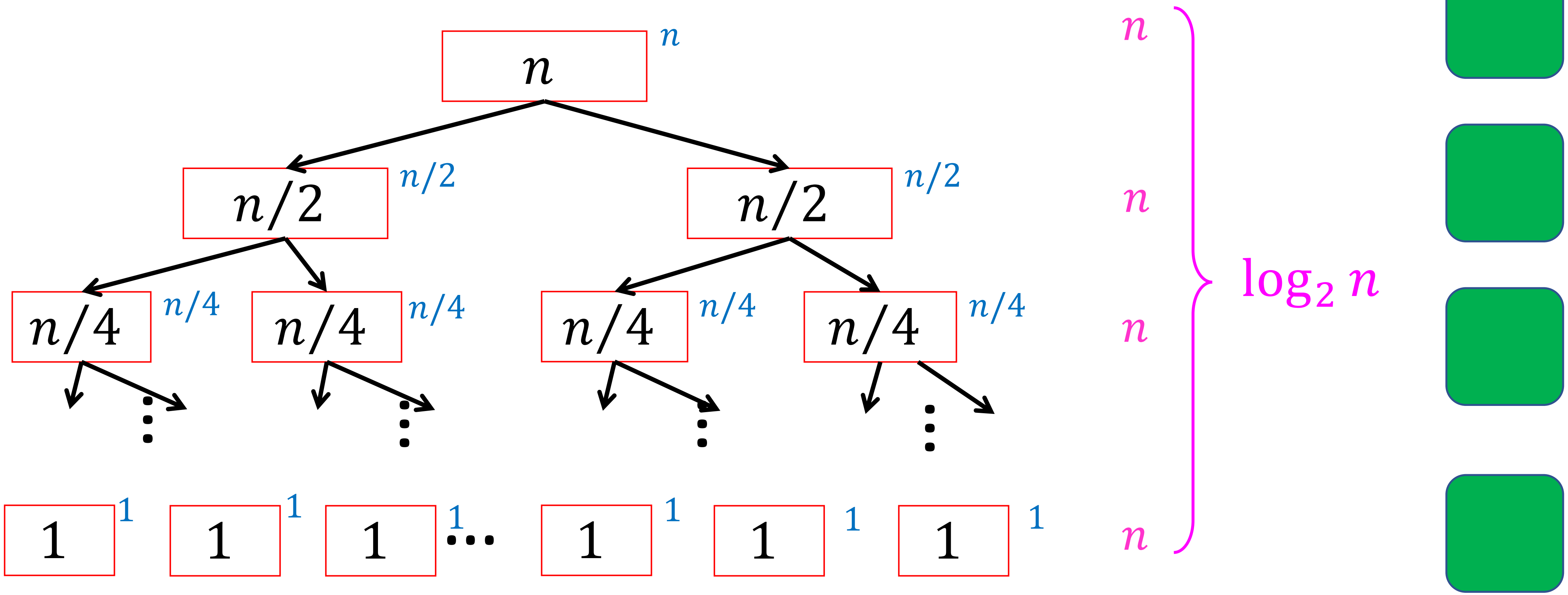
$f(n) = n \in \Theta(n^{\delta})$

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

	Requirement on f	Implication
Case 1	$f(n) \in O(n^{\delta-\varepsilon})$ for some constant $\varepsilon > 0$	$T(n) \in \Theta(n^{\delta})$
Case 2	$f(n) \in \Theta(n^{\delta})$	$T(n) \in \Theta(n^{\delta} \log n)$
Case 3	$f(n) \in \Omega(n^{\delta+\varepsilon})$ for some constant $\varepsilon > 0$ **AND** $af\left(\frac{n}{b}\right) \leq cf(n)$ for constant $c < 1$ and sufficiently large n	$T(n) \in \Theta(f(n))$

Master Theorem Example 1 (Visually)

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



Cost is <u>consistent</u> across levels ⇒

Cost increases by log factor (≈ number of levels)

Master Theorem Example 2

$$T(n) = 4T(n/2) + 5n$$

$T(n) = aT(n/b) + f(n)$ $\delta = \log_b a$

Master Theorem Example 2

$$T(n) = 4T(n/2) + 5n$$

Step 1: Compute $\delta = \log_b a = \log_2 4 = 2$

Step 2: Compare n^δ and $f(n)$

$$f(n) = 5n \in O(n^{2-1}) = O\left(n^{\delta-1}\right)$$

Step 3: Check table

$T(n) = aT(n/b) + f(n)$ $\qquad$ $\delta = \log_b a$

Master Theorem Example 2

$$T(n) = 4T(n/2) + 5n$$

$\delta = 2$

$f(n) = 5n \in O\left(n^{\delta-1}\right)$

	Requirement on f	Implication
Case 1	$f(n) \in O\left(n^{\delta-\varepsilon}\right)$ for some constant $\varepsilon > 0$	$T(n) \in \Theta\left(n^{\delta}\right)$
Case 2	$f(n) \in \Theta\left(n^{\delta}\right)$	$T(n) \in \Theta\left(n^{\delta} \log n\right)$
Case 3	$f(n) \in \Omega\left(n^{\delta+\varepsilon}\right)$ for some constant $\varepsilon > 0$ **AND** $af\left(\frac{n}{b}\right) \leq cf(n)$ for constant $c < 1$ and sufficiently large n	$T(n) \in \Theta\left(f(n)\right)$

Master Theorem Example 2

$$T(n) = 4T(n/2) + 5n$$

$\delta = 2$

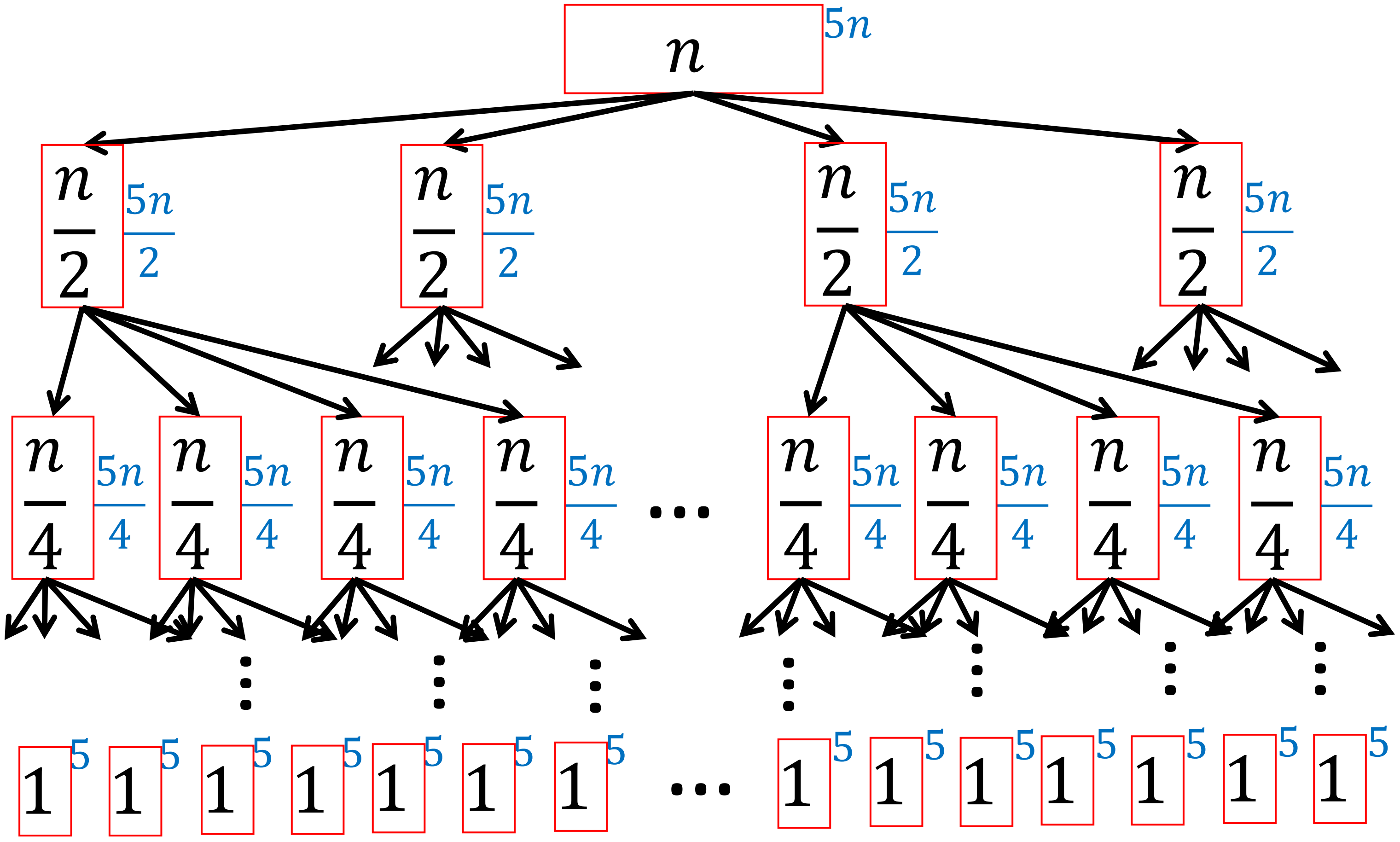
$f(n) = 5n \in O\left(n^{\delta - 1}\right)$

$T(n) = \Theta(n^2)$

	Requirement on f	Implication
Case 1	$f(n) \in O\left(n^{\delta-\varepsilon}\right)$ for some constant $\varepsilon > 0$	$T(n) \in \Theta\left(n^{\delta}\right)$
Case 2	$f(n) \in \Theta\left(n^{\delta}\right)$	$T(n) \in \Theta\left(n^{\delta} \log n\right)$
Case 3	$f(n) \in \Omega\left(n^{\delta+\varepsilon}\right)$ for some constant $\varepsilon > 0$ **AND** $af\left(\frac{n}{b}\right) \leq cf(n)$ for constant $c < 1$ and sufficiently large n	$T(n) \in \Theta(f(n))$

Master Theorem Example 2 (Visually)

$$T(n) = 4T(n/2) + 5n$$



$$5n$$

$$\frac{4}{2} \cdot 5n$$

$$\frac{16}{4} \cdot 5n$$

$$\vdots$$

$$2^{\log_2 n} \cdot 5n$$

Master Theorem Example 2 (Visually)

$$T(n) = 4T(n/2) + 5n$$

Cost is <u>increasing</u> with the recursion depth (due to large number of subproblems)

Most of the work happening in the leaves

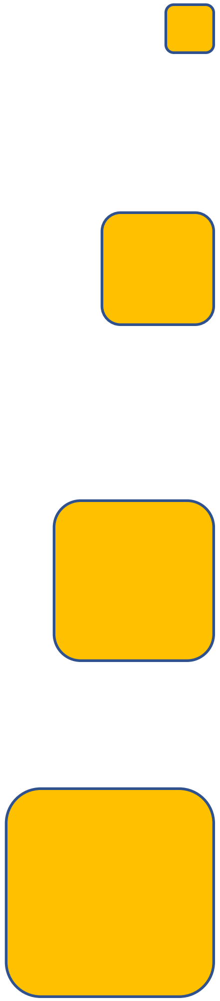
$$5n$$

$$\frac{4}{2} \cdot 5n$$

$$\frac{16}{4} \cdot 5n$$

$$\vdots$$

$$2^{\log_2 n} \cdot 5n$$



Master Theorem Example 3

$$T(n) = 3T(n/2) + 8n$$

[Karatsuba]

$$T(n) = aT(n/b) + f(n)$$

$$\delta = \log_b a$$

Master Theorem Example 3

$$T(n) = 3T(n/2) + 8n$$ [Karatsuba]

Step 1: Compute $\delta = \log_b a = \log_2 3$

Step 2: Compare n^δ and $f(n)$

$f(n) = 8n \in O\left(n^{\log_2 3 - \varepsilon}\right)$ for constant $\varepsilon > \log_2 3 - 1 > 0$

Step 3: Check table

$T(n) = aT(n/b) + f(n)$ $\delta = \log_b a$

Master Theorem Example 3

$T(n) = 3T(n/2) + 8n$ [Karatsuba]

$\delta = \log_2 3$

$f(n) = 5n \in O\left(n^{\delta-\varepsilon}\right)$

	Requirement on f	Implication
Case 1	$f(n) \in O\left(n^{\delta-\varepsilon}\right)$ for some constant $\varepsilon > 0$	$T(n) \in \Theta\left(n^{\delta}\right)$
Case 2	$f(n) \in \Theta\left(n^{\delta}\right)$	$T(n) \in \Theta\left(n^{\delta}\log n\right)$
Case 3	$f(n) \in \Omega\left(n^{\delta+\varepsilon}\right)$ for some constant $\varepsilon > 0$ **AND** $af\left(\frac{n}{b}\right) \leq cf(n)$ for constant $c < 1$ and sufficiently large n	$T(n) \in \Theta\left(f(n)\right)$

Master Theorem Example 3

$T(n) = 3T(n/2) + 8n$ [Karatsuba]

$\delta = \log_2 3$

$f(n) = 5n \in O\left(n^{\delta-\varepsilon}\right)$

$T(n) = \Theta\left(n^{\log_2 3}\right)$

	Requirement on f	Implication
Case 1	$f(n) \in O\left(n^{\delta-\varepsilon}\right)$ for some constant $\varepsilon > 0$	$T(n) \in \Theta\left(n^{\delta}\right)$
Case 2	$f(n) \in \Theta\left(n^{\delta}\right)$	$T(n) \in \Theta\left(n^{\delta} \log n\right)$
Case 3	$f(n) \in \Omega\left(n^{\delta+\varepsilon}\right)$ for some constant $\varepsilon > 0$ **AND** $af\left(\frac{n}{b}\right) \le cf(n)$ for constant $c < 1$ and sufficiently large n	$T(n) \in \Theta\left(f(n)\right)$

Master Theorem Example 4

$$T(n) = 2T(n/2) + 15n^3$$

$T(n) = aT(n/b) + f(n)$ $\delta = \log_b a$

Master Theorem Example 4

$$T(n) = 2T(n/2) + 15n^3$$

Step 1: Compute $\delta = \log_b a = \log_2 2 = 1$

Step 2: Compare n^δ and $f(n)$

$$f(n) = 15n^3 \in \Omega(n^{1+2}) = \Omega\left(n^{\delta+2}\right)$$

Step 3: Check table

$T(n) = aT(n/b) + f(n)$ $\qquad$ $\delta = \log_b a$

Master Theorem Example 4

$$T(n) = 2T(n/2) + 15n^3$$

$\delta = 1$

$f(n) = 15n^3 \in \Omega\left(n^{\delta+2}\right)$

	Requirement on f	Implication
Case 1	$f(n) \in O\left(n^{\delta-\varepsilon}\right)$ for some constant $\varepsilon > 0$	$T(n) \in \Theta\left(n^{\delta}\right)$
Case 2	$f(n) \in \Theta\left(n^{\delta}\right)$	$T(n) \in \Theta\left(n^{\delta} \log n\right)$
Case 3	$f(n) \in \Omega\left(n^{\delta+\varepsilon}\right)$ for some constant $\varepsilon > 0$ **AND** $af\left(\frac{n}{b}\right) \leq cf(n)$ for constant $c < 1$ and sufficiently large n	$T(n) \in \Theta\left(f(n)\right)$

Master Theorem Example 4

$$T(n) = 2T(n/2) + 15n^3$$

$\delta = 1$

$f(n) = 15n^3 \in \Omega\left(n^{\delta+2}\right)$

	Requirement on f	Implication
Case 1	$f(n) \in O\left(n^{\delta-\varepsilon}\right)$ for some constant $\varepsilon > 0$	$T(n) \in \Theta\left(n^{\delta}\right)$
Case 2	$f(n) \in \Theta\left(n^{\delta}\right)$	$T(n) \in \Theta\left(n^{\delta} \log n\right)$
Case 3	$f(n) \in \Omega\left(n^{\delta+\varepsilon}\right)$ for some constant $\varepsilon > 0$ **AND** $af\left(\frac{n}{b}\right) \leq cf(n)$ for constant $c < 1$ and sufficiently large n	$T(n) \in \Theta\left(f(n)\right)$

Master Theorem Example 4

$$T(n) = 2T(n/2) + 15n^3$$

$\delta = 1$

$f(n) = 15n^3 \in \Omega\left(n^{\delta+2}\right)$

Important: For Case 3, need to additionally check that $2f(n/2) \leq cf(n)$ for constant $c < 1$ and sufficiently large n

$$2f(n/2) = 30(n/2)^3 = \frac{30}{8}n^3 \leq \frac{1}{4}(15n^3)$$

Master Theorem Example 4

$T(n) = 2T(n/2) + 15n^3$

$\delta = 1$

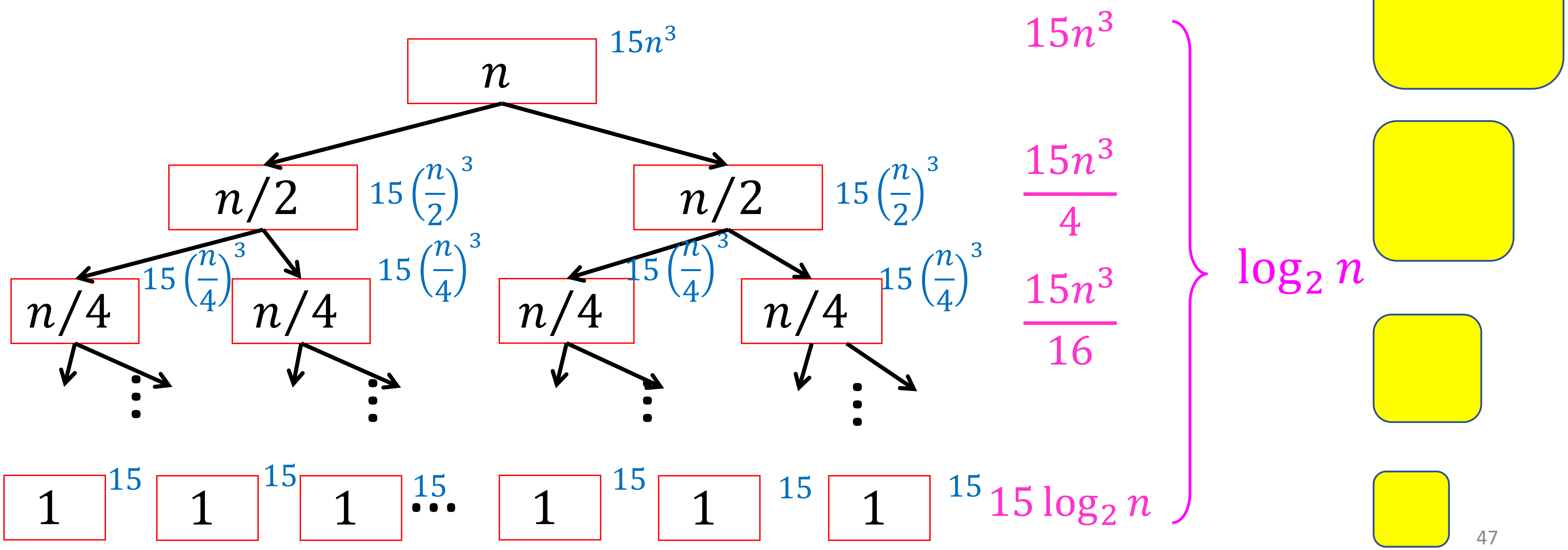
$f(n) = 15n^3 \in \Omega\left(n^{\delta+2}\right)$

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

	Requirement on f	Implication
Case 1	$f(n) \in O\left(n^{\delta-\varepsilon}\right)$ for some constant $\varepsilon > 0$	$T(n) \in \Theta\left(n^{\delta}\right)$
Case 2	$f(n) \in \Theta\left(n^{\delta}\right)$	$T(n) \in \Theta\left(n^{\delta}\log n\right)$
Case 3	$f(n) \in \Omega\left(n^{\delta+\varepsilon}\right)$ for some constant $\varepsilon > 0$ **AND** $af\left(\frac{n}{b}\right) \leq cf(n)$ for constant $c < 1$ and sufficiently large n	$T(n) \in \Theta\left(f(n)\right)$

Master Theorem Example 3 (Visually)

$$T(n) = 2T(n/2) + 15n^3$$



Master Theorem Example 3 (Visually)

$$T(n) = 2T(n/2) + 15n^3$$

Cost is decreasing with the recursion depth (due to high *non-recursive* cost)

Most of the work happening at the top

$15n^3$

$\frac{15n^3}{4}$

$\frac{15n^3}{16}$

$15 \log_2 n$

$\log_2 n$