CS 4102: Algorithms

Lecture 5: Master Theorem

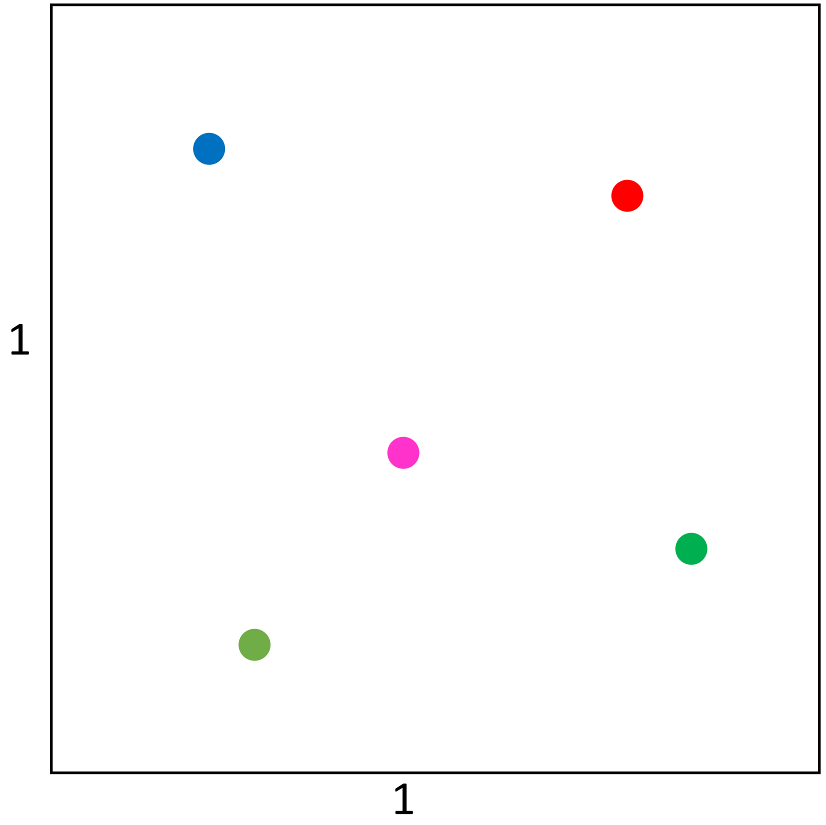
David Wu

Fall 2019

Warm Up

Warm up

Given any 5 points on the unit square, show there's always a pair distance $\leq \frac{\sqrt{2}}{2}$ apart

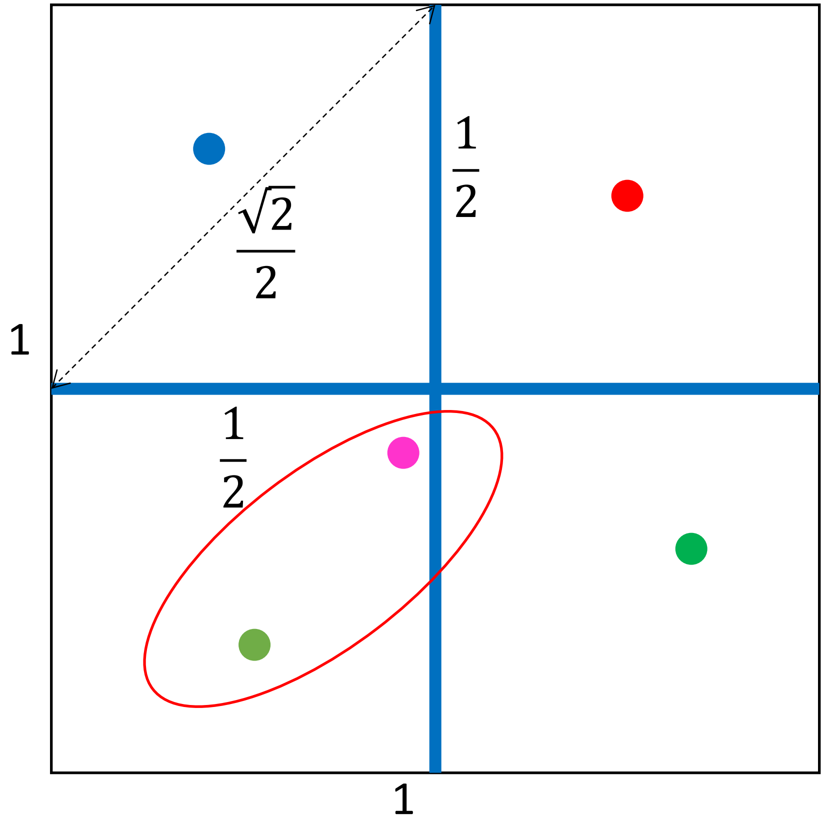


Warm Up

If points p_1, p_2 in same quadrant, then $d(p_1, p_2) \leq \frac{\sqrt{2}}{2}$

Given 5 points, two must share the same quadrant

Pigeonhole Principle!



Today's Keywords

Solving recurrences

Cookbook Method

Master Theorem

Substitution Method

CLRS Readings: Chapter 4

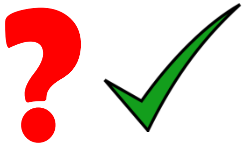
Homework

HW1 due ~~**Thursday, September 12**~~ **Saturday, September 14**, **11pm**

- Start early!
- Written (use Latex!) – Submit both **zip** and **pdf**!
- Asymptotic notation
- Recurrences
- Divide and Conquer

Recurrence Solving Techniques

Tree



Guess/Check (induction)

"Cookbook"

Substitution

Induction (Review)

Goal:	$\forall k \in \mathbb{N}, P(k)$ holds
Base case(s):	$P(1)$ holds
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)

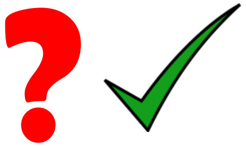
Technique: Induction

- 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

Recurrence Solving Techniques

Tree



Guess/Check

"Cookbook"

Substitution

Observation

Divide: $D(n)$ time

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

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

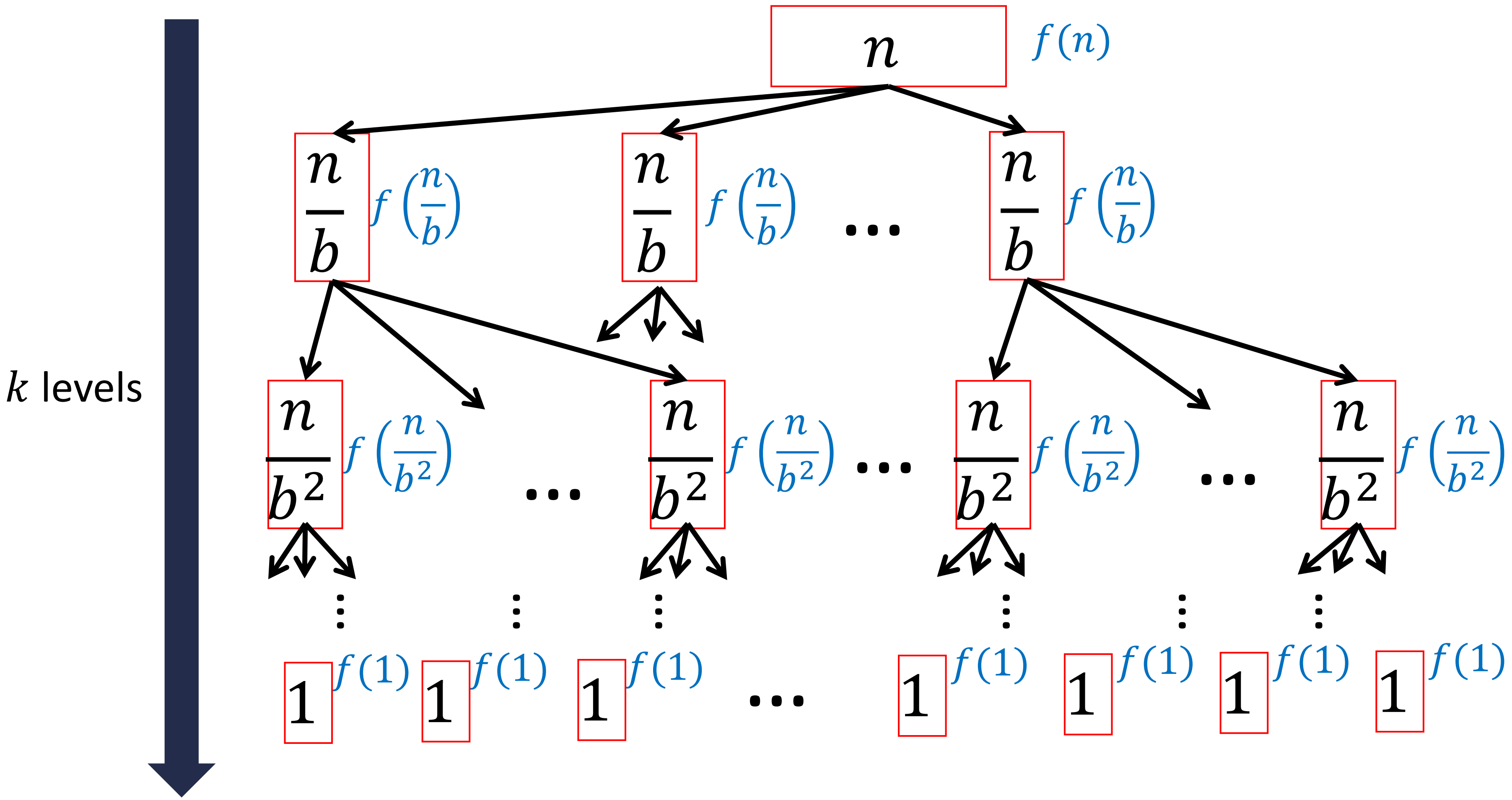
Mergesort: $T(n) = 2T(n/2) + n$

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

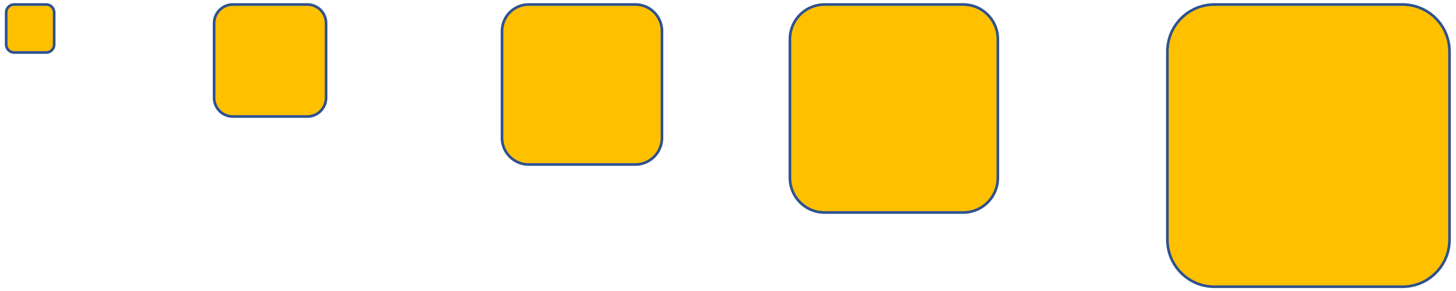
Number of subproblems	Cost of subproblem
1	$f(n)$
a	$f(n/b)$
a^2	$f(n/b^2)$
a^k	$f(n/b^k)$

Three Cases

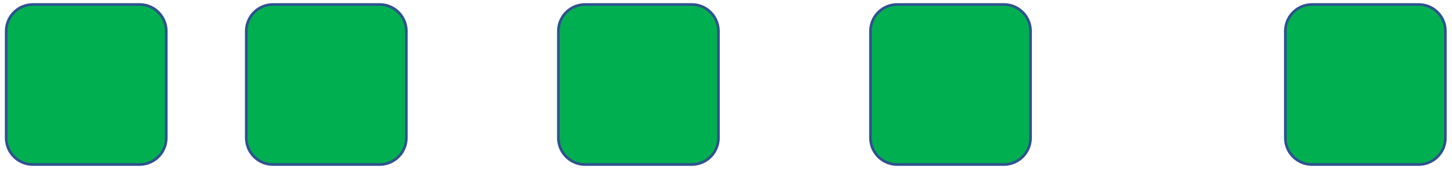
$$T(n) = f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^k f\left(\frac{n}{b^k}\right)$$

$$k = \log_b n$$

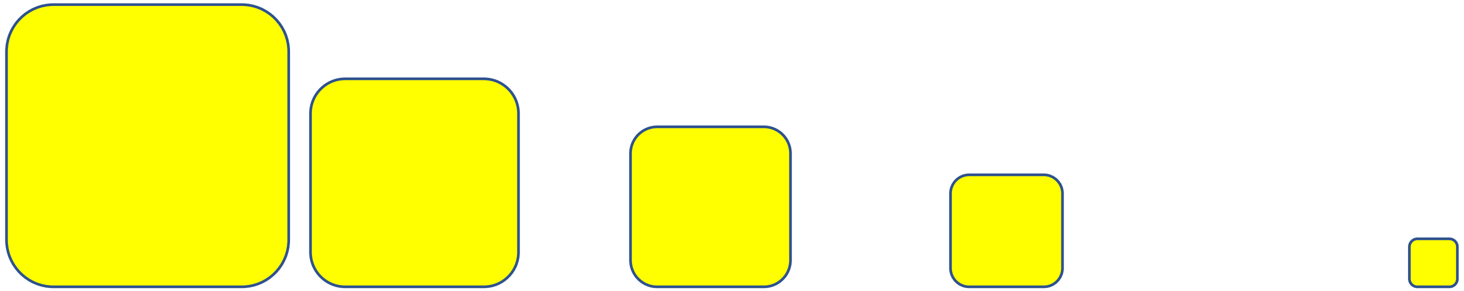
Case 1:
Most work happens
at the leaves



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	$f(n) \in O(n^{\delta-\varepsilon})$ for some constant $\varepsilon > 0$	$T(n) \in \Theta(n^{\delta})$

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	$T(n) \in \Theta\left(f(n)\right)$

Proof of Case 1

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

We will show weaker version of Case 1:

if $f(n) \in O(n^{\delta-\varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) \in O(n^{\delta}) = O(n^{\log_b a})$

There exists constants c, n_0 such that for all $n > n_0$, $f(n) \leq cn^{\delta-\varepsilon}$

Similar argument applies to show that $T(n) = \Omega(n^{\log_b a})$

We will consider $n \geq n_0$

Proof of Case 1

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

We will show weaker version of Case 1:

if $f(n) \in O\left(n^{\delta-\varepsilon}\right)$ for some constant $\varepsilon > 0$, then $T(n) \in O\left(n^{\delta}\right) = O\left(n^{\log_b a}\right)$

$$\begin{aligned} T(n) &= f(n) + af\left(\frac{n}{b}\right) + a^2 f\left(\frac{n}{b^2}\right) + a^3 f\left(\frac{n}{b^3}\right) + \cdots + a^k f\left(\frac{n}{b^k}\right) \\ &\le c\left(n^{\delta-\varepsilon} + a\left(\frac{n}{b}\right)^{\delta-\varepsilon} + a^2\left(\frac{n}{b^2}\right)^{\delta-\varepsilon} + \cdots + a^k\left(\frac{n}{b^k}\right)^{\delta-\varepsilon}\right) \qquad n > n_0 \Rightarrow f(n) \le cn^{\delta-\varepsilon} \\ &= cn^{\delta-\varepsilon}\left(1 + a\left(\frac{1}{b}\right)^{\delta-\varepsilon} + a^2\left(\frac{1}{b^2}\right)^{\delta-\varepsilon} + \cdots + a^k\left(\frac{1}{b^k}\right)^{\delta-\varepsilon}\right) \end{aligned}$$

Proof of Case 1

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

$$\delta = \log_b a$$

$$T(n) \le cn^{\delta-\varepsilon}\left(1 + a\left(\frac{1}{b}\right)^{\delta-\varepsilon} + a^2\left(\frac{1}{b^2}\right)^{\delta-\varepsilon} + \cdots + a^k\left(\frac{1}{b^k}\right)^{\delta-\varepsilon}\right)$$

$$= cn^{\delta-\varepsilon}\left(1 + \left(\frac{a}{b^{\delta-\varepsilon}}\right) + \left(\frac{a}{b^{\delta-\varepsilon}}\right)^2 + \cdots + \left(\frac{a}{b^{\delta-\varepsilon}}\right)^k\right)$$

$$= cn^{\delta-\varepsilon}\left(1 + \left(\frac{ab^{\varepsilon}}{b^{\delta}}\right) + \left(\frac{ab^{\varepsilon}}{b^{\delta}}\right)^2 + \cdots + \left(\frac{ab^{\varepsilon}}{b^{\delta}}\right)^k\right)$$

$$= cn^{\delta-\varepsilon}\left(1 + \left(\frac{ab^{\varepsilon}}{a}\right) + \left(\frac{ab^{\varepsilon}}{a}\right)^2 + \cdots + \left(\frac{ab^{\varepsilon}}{a}\right)^k\right)$$

$$b^{\delta} = b^{\log_b a} = a$$

Proof of Case 1

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

$$T(n) \le cn^{\delta-\varepsilon}\left(1 + \left(\frac{ab^{\varepsilon}}{a}\right) + \left(\frac{ab^{\varepsilon}}{a}\right)^2 + \cdots + \left(\frac{ab^{\varepsilon}}{a}\right)^k\right)$$

$$= cn^{\delta-\varepsilon}\left(1 + b^{\varepsilon} + b^{2\varepsilon} + \cdots + b^{k\varepsilon}\right)$$

$$= cn^{\delta-\varepsilon}\left(\sum_{i=0}^{k}(b^{\varepsilon})^i\right)$$

$$= cn^{\delta-\varepsilon}\frac{(b^{\varepsilon})^{k+1} - 1}{b^{\varepsilon} - 1}$$

$$\sum_{i=0}^{L} a^i = \frac{a^{L+1} - 1}{a - 1}$$

Proof of Case 1

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

$$\begin{aligned} T(n) &\le cn^{\delta-\varepsilon}\frac{(b^{\varepsilon})^{k+1}-1}{b^{\varepsilon}-1} \\ &= cn^{\delta-\varepsilon}\frac{\left(b^{k}\right)^{\varepsilon}\cdot b^{\varepsilon}-1}{b^{\varepsilon}-1} \\ &= cn^{\delta-\varepsilon}\frac{n^{\varepsilon}\cdot b^{\varepsilon}-1}{b^{\varepsilon}-1} \\ &= \frac{cb^{\varepsilon}}{b^{\varepsilon}-1}n^{\delta} - \frac{c}{b^{\varepsilon}-1}n^{\delta-\varepsilon} \end{aligned}$$

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

Proof of Case 1

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

$$\delta = \log_b a$$

$$T(n) \leq cn^{\delta-\varepsilon}\frac{(b^\varepsilon)^{k+1} - 1}{b^\varepsilon - 1}$$

$$= cn^{\delta-\varepsilon}\frac{\left(b^k\right)^\varepsilon \cdot b^\varepsilon - 1}{b^\varepsilon - 1}$$

$$= cn^{\delta-\varepsilon}\frac{n^\varepsilon \cdot b^\varepsilon - 1}{b^\varepsilon - 1}$$

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

$$= \frac{cb^\varepsilon}{b^\varepsilon - 1}n^\delta - \frac{c}{b^\varepsilon - 1}n^{\delta-\varepsilon}$$

$$\in O\left(n^\delta\right)$$

b, c, ε are all constants

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

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

$\delta = 1$

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

	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

$\delta = 1$

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

[Merge Sort]

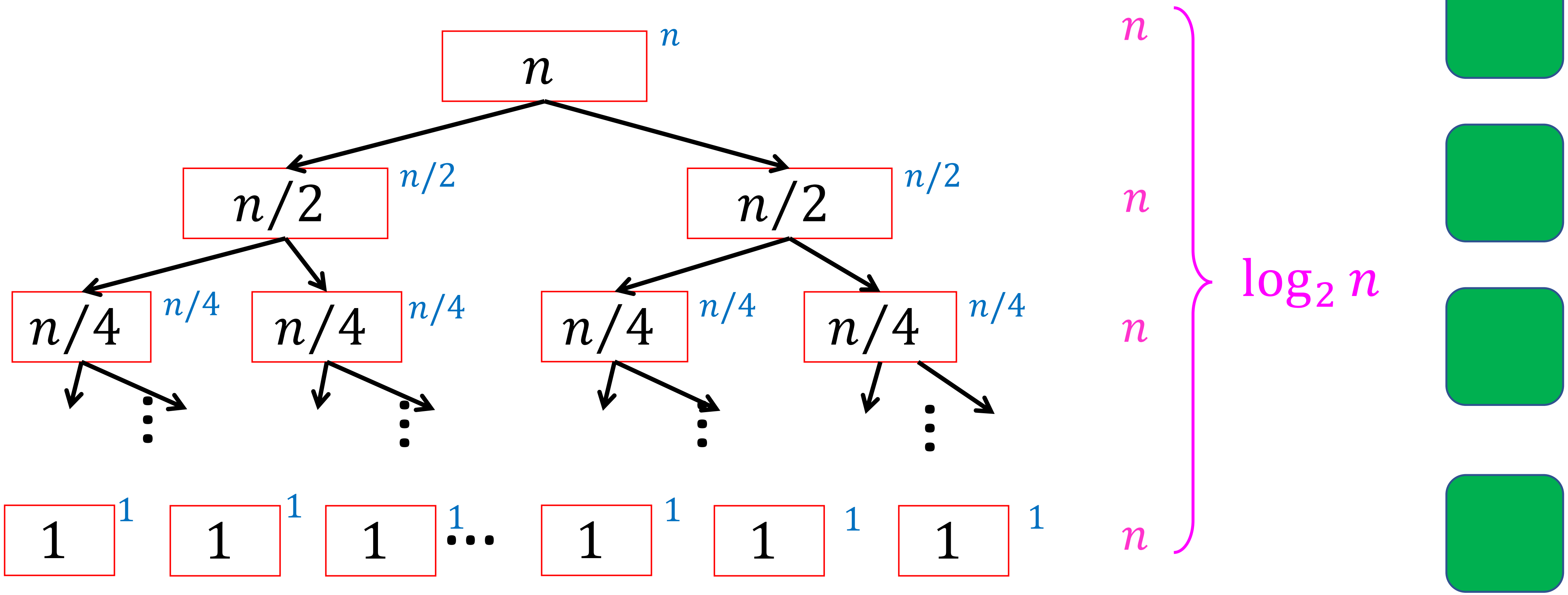
$f(n) = n \in \Theta(n^{\delta})$

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

	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

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

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

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

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

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

$\delta = 2$

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

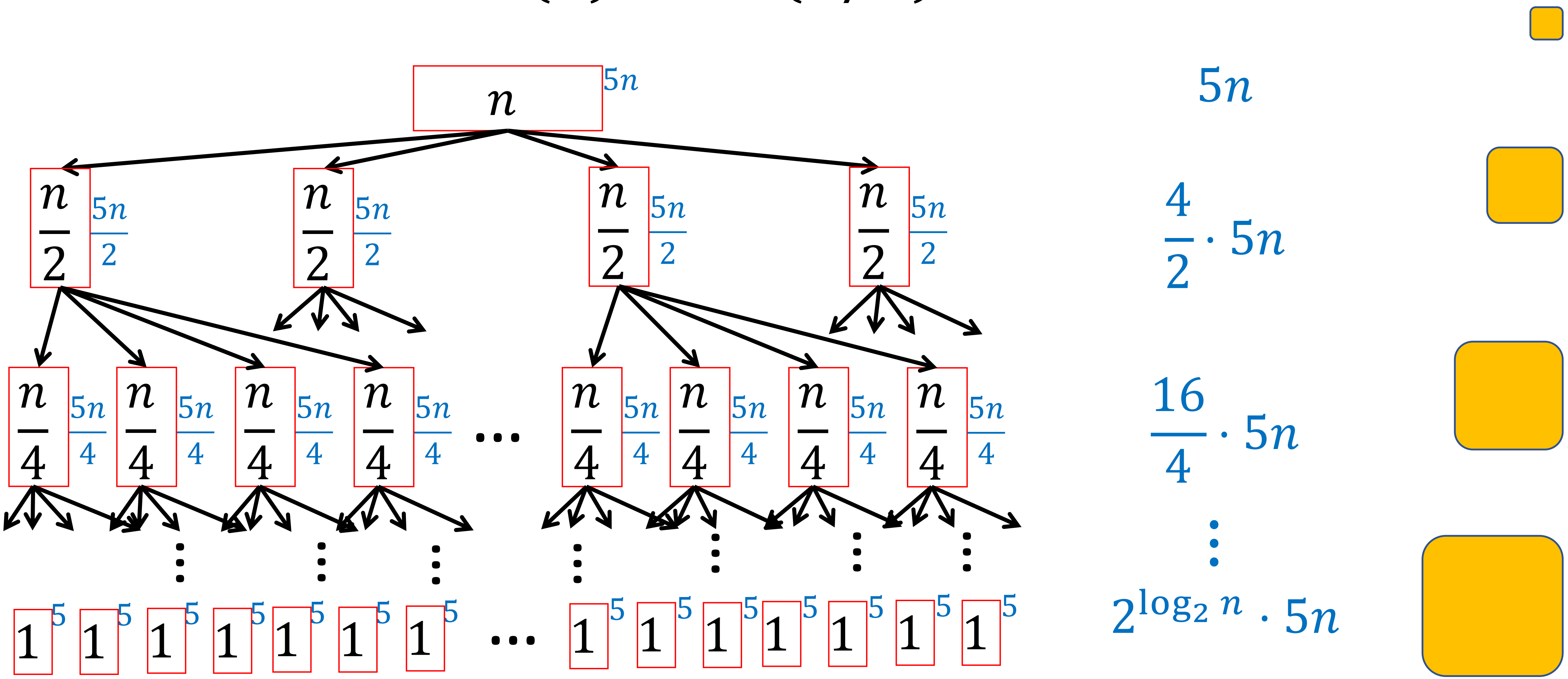
$f(n) = 5n \in O\left(n^{\delta-1}\right)$

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

	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)

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



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

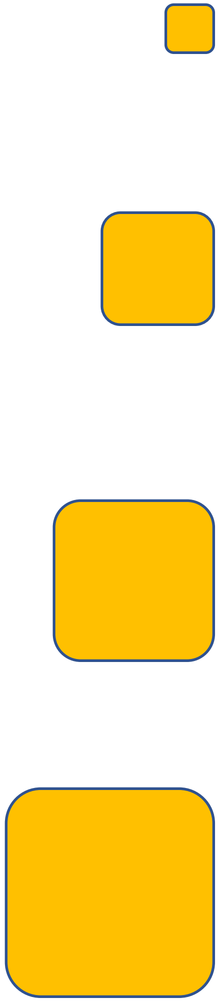
$$5n$$

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

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

$$\vdots$$

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



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 $f(n)$

$f(n) = 5n \in O\left(n^{\log_2 3 - \varepsilon}\right)$ for constant $\varepsilon > \log_2 3 - 1 > 0$

Step 3: Check table

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

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** $af\left(\frac{n}{b}\right) \leq cf(n)$ for constant $c < 1$ and sufficiently large n	$T(n) \in \Theta\left(f(n)\right)$

Master Theorem Example 4

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

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

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 $f(n)$

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

Step 3: Check table

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

Master Theorem Example 4

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

$\delta = 1$

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

	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** $af\left(\frac{n}{b}\right) \le cf(n)$ for constant $c < 1$ and sufficiently large n	$T(n) \in \Theta(f(n))$

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 $c \geq 1$ 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

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

$\delta = 1$

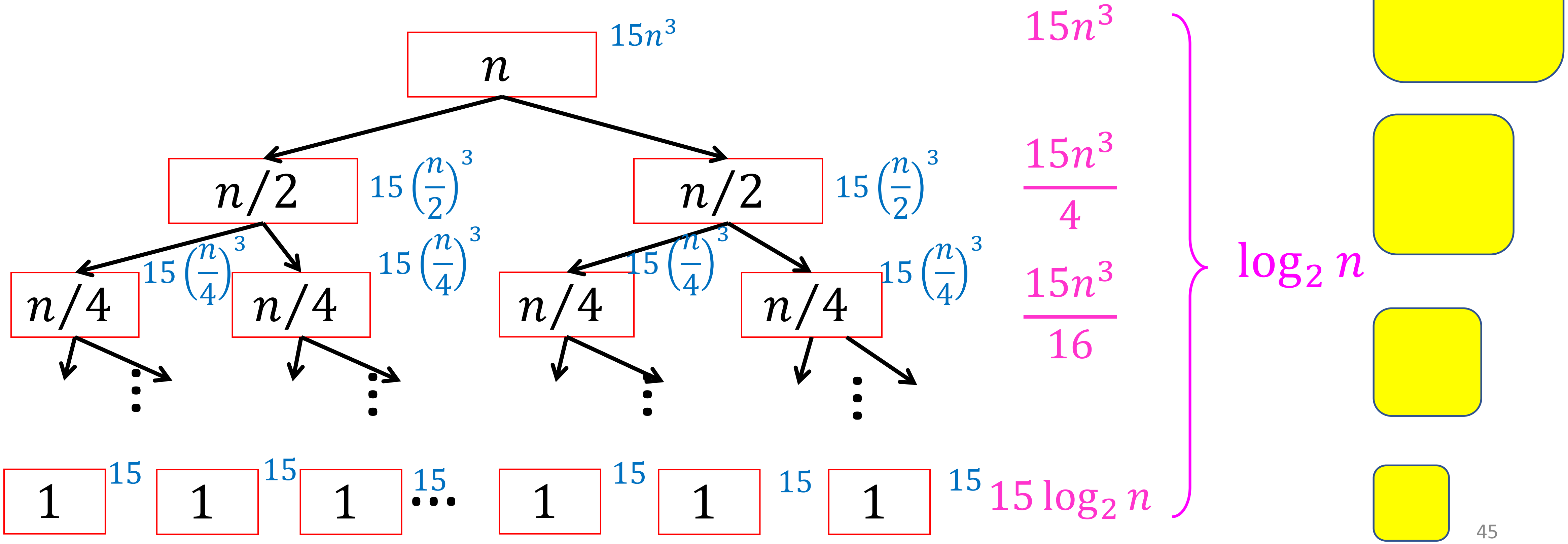
$f(n) = 15n^3 \in \Omega\left(n^{\delta+2}\right)$

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

	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** $af\left(\frac{n}{b}\right) \le cf(n)$ 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

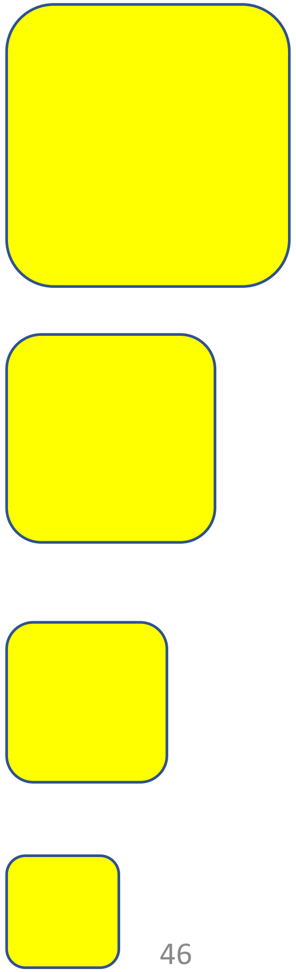
$15n^3$

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

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

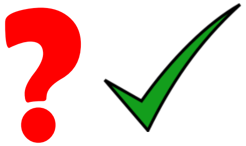
$15 \log_2 n$

$\log_2 n$



Recurrence Solving Techniques

Tree



Guess/Check

"Cookbook"

Substitution

Substitution Method

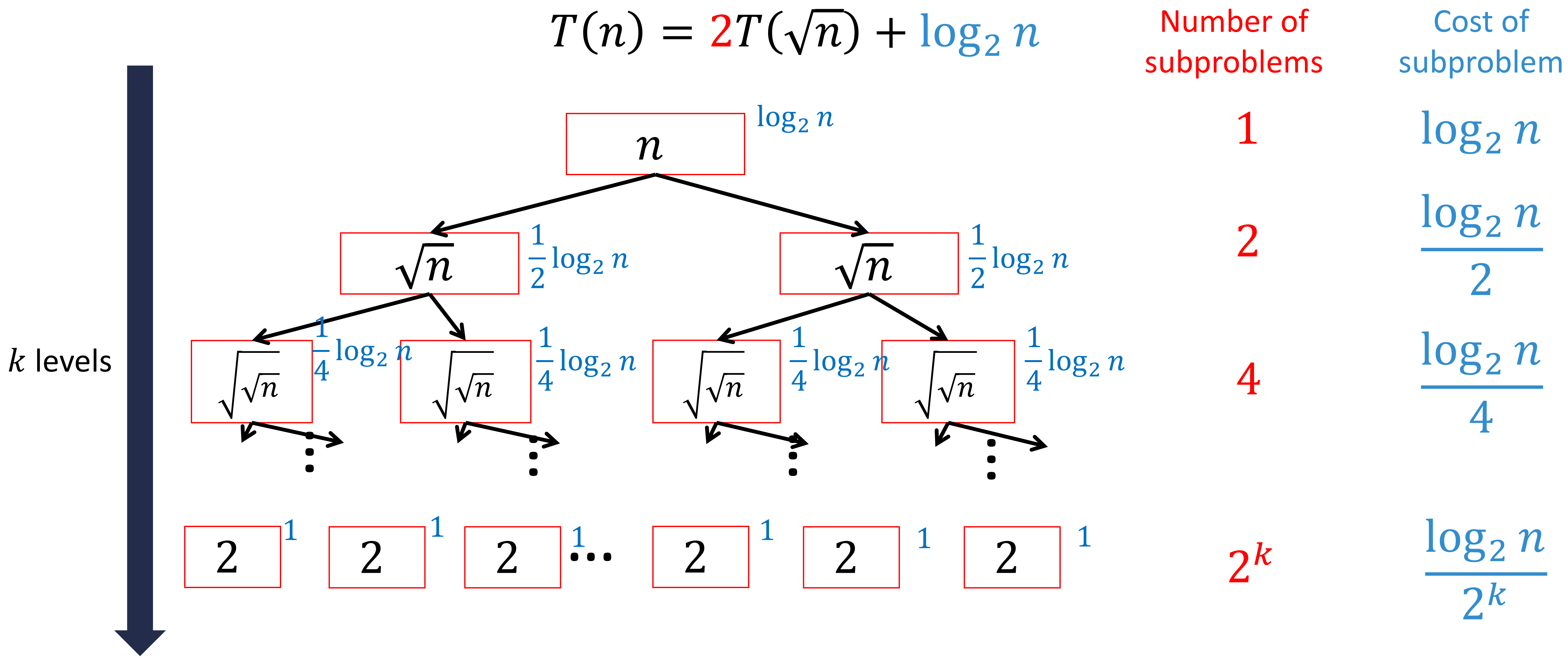
Idea: Take a “difficult” recurrence and re-express it such that one of our other methods applies

Example:

$$T(n) = 2T(\sqrt{n}) + \log_2 n$$

Tree Method

$$T(n) = 2T(\sqrt{n}) + \log_2 n$$



Tree Method

$$T(n) = 2T(\sqrt{n}) + \log_2 n$$

Each iteration, problem size goes from n to $n^{1/2}$

How many levels?

Problem size at k^{th} level: $n^{1/2^k}$

Base case: $n = 2$

At level k, it should be the case that $n^{1/2^k} = 2$

$$n^{1/2^k} = 2 \Rightarrow \frac{1}{2^k}\log_2 n = 1$$

$$\Rightarrow 2^k = \log_2 n \Rightarrow k = \log_2 \log_2 n$$

Number of subproblems	Cost of subproblem
1	$\log_2 n$
2	$\frac{\log_2 n}{2}$
4	$\frac{\log_2 n}{4}$
2^k	$\frac{\log_2 n}{2^k}$

Tree Method

$$T(n) = 2T(\sqrt{n}) + \log_2 n$$

	Number of subproblems	Cost of subproblem
	1	$\log_2 n$
	2	$\frac{\log_2 n}{2}$
	4	$\frac{\log_2 n}{4}$
	2^k	$\frac{\log_2 n}{2^k}$

$$k = \log_2 \log_2 n$$

What is the cost?

Cost at level i: $2^i \cdot \frac{\log_2 n}{2^i} = \log_2 n$

Total cost: $T(n) = \sum_{i=0}^{\log_2 \log_2 n} \log_2 n = \log_2 n \log_2 \log_2 n$

Substitution Method

Idea: Take a "difficult" recurrence and re-express it such that one of our other methods applies

Example:

$$T(n) = 2T(\sqrt{n}) + \log_2 n$$

Consider the following substitution: let $n = 2^m$ (i.e., $m = \log_2 n$)

$$T(2^m) = 2T\left(2^{\frac{m}{2}}\right) + m$$ Rewrite recurrence in terms of m

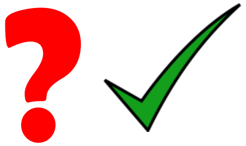
$$S(m) = 2S\left(\frac{m}{2}\right) + m$$ Consider substitution $S(m) = T(2^m)$

$$\Rightarrow S(m) = \Theta(m \log m)$$ Case 2 of Master Theorem

$$\Rightarrow T(n) = \Theta(\log n \log\log n)$$ Substitute back for T and n

Recurrence Solving Techniques

Tree



Guess/Check

"Cookbook"

Substitution