6.046
Lecture 2

Recurrences, Substitution, Master Method

Assymptotic notation

O notation ($\leq$)

$f(n) = O(g(n))$ means $\exists\ c>0, n_0>0,$
s.t. $0 \leq f(n) \leq c\,g(n)\ \ \forall n \geq n_0$

asymmetric due to $\in$ — notation

Ex: $2n^2 = O(n^3)$, $2n^2 \in O(n^3)$ → 2 equiv. ways

Set definition:

$O(g(n)) = \{f(n) : \exists\ c>0, n_0>0, \text{ s.t. } 0 \leq f(n) \leq c\,g(n), \forall n \geq n_0\}$

Macro convention:

A set in a formula represents an anonymous function in that set.

Ex: $f(n) = n^3 + O(n^2)$
means $\exists\ h(n) \in O(n^2)$, s.t. $f(n) = n^3 + h(n)$

Ex: $n^2 + O(n) = O(n^2)$ [redacted]

means: $\forall f(n) \in O(n)$
$\exists\ h(n) \in O(n^2)$, s.t. $n^2 + f(n) = h(n)$

Ω notation: ($\geq$)

$\Omega(g(n)) = \{f(n) : \exists\ c>0, n_0>0, \text{ s.t. } 0 \leq c\,g(n) \leq f(n), \forall n \geq n_0\}$

Ex: $\sqrt{n} = \Omega(\lg n)$ $\sqrt{n} \neq O(\lg n)$ $\sqrt{n} \neq \Theta(\lg(n))$

[redacted]

$\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$ ($=$)

Analogies

O	Ω	Θ	o	ω
$\leq$	$\geq$	$=$	$<$	$>$

o & ω notations

inequality must hold $\forall c>0$

Ex: $2n^2 = o(n^3)$ $\left(n_0 = \frac{2}{c}\right)$

$\frac{1}{2}n^2 = \Theta(n^2)$
$\neq o(n^2)$

Solving Recurrences

Substitution method

1. guess the form of the solution
2. verify by induction
3. solve for consts.

Cannot induce on $O(n)$
$n = O(1)$, $1 = O(1)$, $n-1 = O(1)$ NOT!
$\Rightarrow n = (n-1)+1 = O(1)$

Ex 1.

$$T(n) = 4T\left(\frac{n}{2}\right) + n$$

base $\to T(1) = \Theta(1) \le c_1$ — true if c is suf. large

$T(2) = 4\,\Theta(1) + 2$, $T(2) \le c4+2 \le c2^3$, $c \ge 1/2$

& prove
- given $T(n) = O(n^3)$
- assume $T(k) \le ck^3$ for $k < n$ (IH)
- $T(n) = 4T\left(\frac{n}{2}\right) + n$

$\le 4c\left(\frac{n}{2}\right)^3 + n$ by IH

$= \frac{1}{2}cn^3 + n$

$= \underbrace{cn^3}_{\text{desired}} - \underbrace{\left(\frac{1}{2}cn^3 - n\right)}_{\text{residual}}$

$\le cn^3$, if $\frac{1}{2}cn^3 - n \ge 0$

e.g. $c = 2$ $(n \ge$ [redacted]

Pick c at the end of induction

[redacted]

Ex 2

$$T(n) = 4T\left(\frac{n}{2}\right) + n$$

prove: $T(n) = O(n^2)$

assume $T(k) \le ck^2$, $k < n$

$T(n) = 4T\left(\frac{n}{2}\right) + n$

$\le cn^2 + n$ by IH

$= cn^2 - \underbrace{(-n)}$ ← want ≥ 0

$\not\le cn^2$ $\quad n \ge 1$

Fix: stronger IH

assume $T(k) \le c_1k^2 - c_2k$ for $k < n$

$T(n) = 4T\left(\frac{n}{2}\right) + n$

$\le 4\left[c_1\left(\frac{n}{2}\right)^2 - c_2\left(\frac{n}{2}\right)\right] + n$

$= c_1n^2 + (1 - 2c_2)n$

$= \underbrace{c_1n^2 - c_2n}_{\text{desired}} - \underbrace{(-1+c_2)n}_{\text{residual}}$ residual ≥ 0 if $c_2 \ge 1$

$\le c_1n^2 - c_2n$ if $c_2 \ge 1$

base

$T(1) = c_1 - c_2$ $\qquad c_1 > c_2$ [redacted] if

$T(1) = \Theta(1)$ $\qquad c_1$ is sufficiently large with respect to c_2

Prove that induction works for a choice of constants, same across base case and the inductive step.

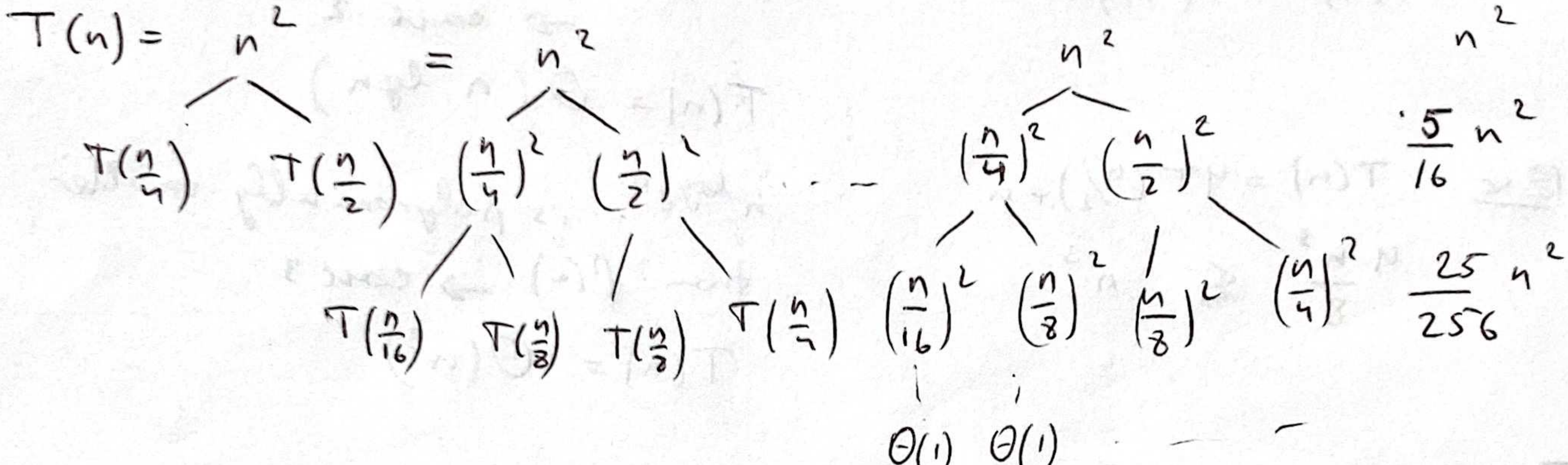
6.046
Lecture 2

(2)

Recursion-tree method

Ex $T(n) = T(\frac{n}{4}) + T(\frac{n}{2}) + n^2$

geometric series



Total (level by level) # of leaves $< n$

$\leq \left(1 + \frac{5}{16} + \frac{25}{256} + \dots + \left(\frac{5}{16}\right)^k + \dots\right) n^2$ $\qquad 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots = 2$

$< 2n^2$ $\qquad T(n) = O(n^2)$

restriction: each sub-problem must be of the same size

does not apply to above rec.

Master method

applies to recurrences of the form $T(n) = a\,T(\frac{n}{b}) + f(n)$

where, $a \geq 1$, $b > 1$, $f(n)$ is asymptotically positive

$f(n) > 0$ for $n \geq n_0$

- compare $f(n)$ with $n^{\log_b a}$

Case 1 $f(n) = O(n^{\log_b a - \varepsilon})$ for some $\varepsilon > 0$

$T(n) = \Theta(n^{\log_b a})$

$f(n)$ must be polynomially smaller than $n^{\log_b a}$

Case 2 $f(n) = \Theta(n^{\log_b a} \lg^k n)$ for some $k \geq 0$

$T(n) = \Theta(n^{\log_b a} \lg^{k+1} n)$

$f(n)$ is pretty much equal to $n^{\log_b a}$, upto poly lg factors

$\lg^k n = (\log_2 n)^k$

Case 3 $f(n) = \Omega(n^{\log_b a + \varepsilon})$ for some $\varepsilon > 0$

& $a f(\frac{n}{b}) \leq (1 - \varepsilon') \cdot f(n)$ for some $\varepsilon' > 0$

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

$f(n)$ is bigger polynomially

make sure $f(n)$ get smaller down the recursion by a const. factor

Ex: $T(n) = \underbrace{4}_{a}T(n/\underbrace{2}_{b}) + \underbrace{n}_{f(n)}$

$n^{\log_b a} = n^2$, bigger than $f(n)$ by a polynomial factor $\to$ case 1

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

Ex: $T(n) = 4T(n/2) + n^2$

n^2 is asymptotically equal to n^2 $\to$ case 2

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

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

$4\frac{n^3}{8} \leq n^3$

$n^{\log_b a}$ is polynomially smaller than $f(n)$ $\to$ case 3

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

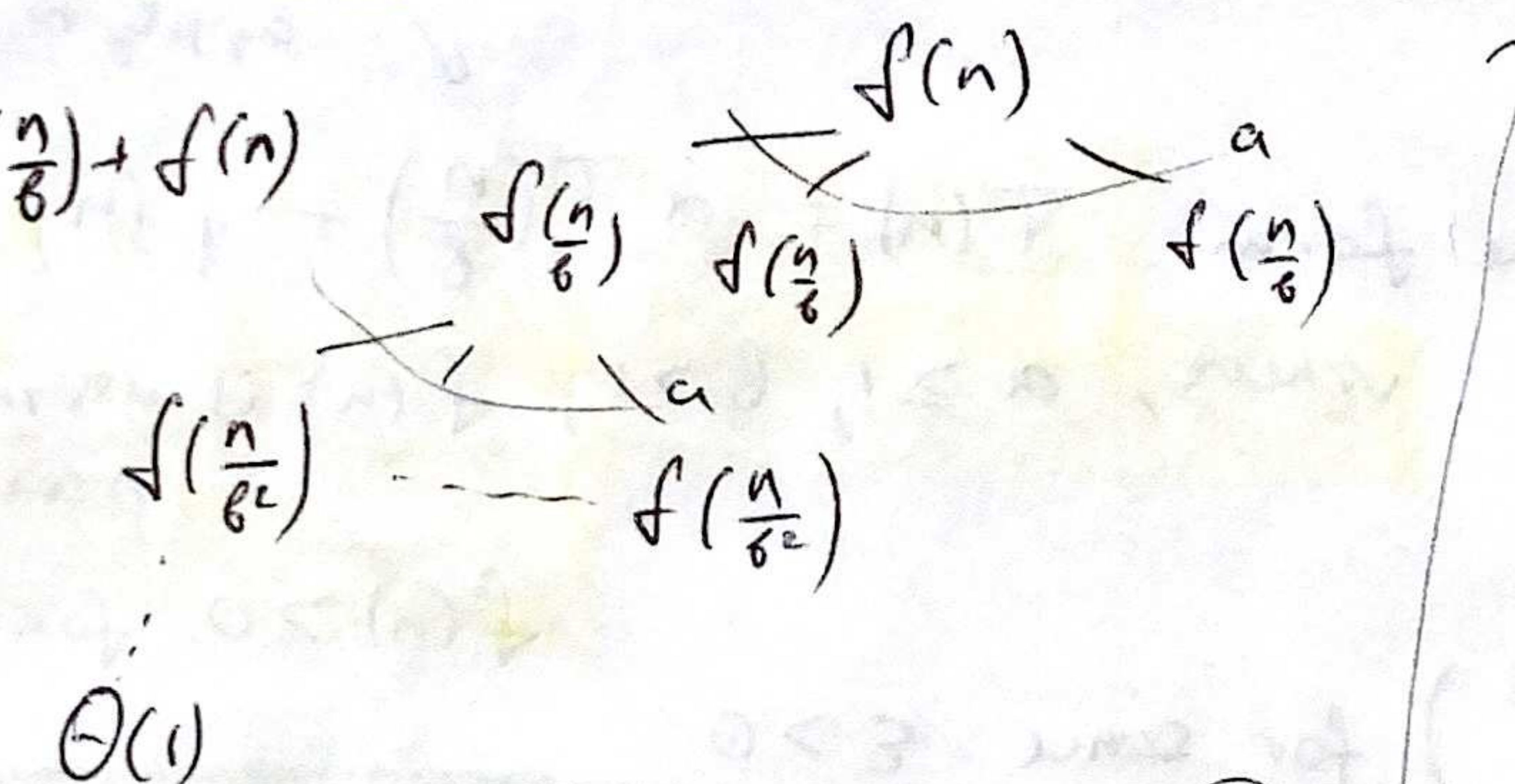
Ex: $T(n) = 4T(n/2) + n^2/\lg n = 4T(n/2) + n^2 \lg^{-1} n$

↑ does not follow from the master method

– use recursion tree

Proof sketch behind the master method.

$T(n) = aT(\frac{n}{b}) + f(n)$



\# leaves $a^h = a^{\log_b n} = n^{\log_b a}$

computation cost

$\leftarrow f(n)$

$\leftarrow a f(\frac{n}{b})$

$\leftarrow a^2 f(\frac{n}{b^2})$

$\vdots$

$\leftarrow \Theta(n^{\log_b a})$

case 2: each level is roughly the same

cost $= f(n) \cdot h$
$\quad\quad\quad\quad\uparrow \Theta(\lg n)$

$\to$ [redacted] $\Theta(n^{\log_b a} \lg^{k+1} n)$

case 1: $f(n)$ is polynomially smaller than [redacted] $n^{\log_b a}$

$\to$ sequence increases geometrically (decreases from bottom to top)

$\to$ geometric series where $n^{\log_b a}$ dominates

$\to \Theta(n^{\log_b a})$

case 3: sequence decreases geometrically

$\to$ upper term dominant

$\to \Theta(f(n))$