CS3230 Tutorial 2

Recurrences and Master Theorem

(AY 25/26 Semester 2)

January 26, 2026

(Prepared by Benson)

Contents

Recap: Telescoping

$$\frac{T(n)}{g(n)} = \frac{T(\frac{n}{b})}{g(\frac{n}{b})} + h(n)$$

$$\frac{T(\frac{n}{b})}{g(\frac{n}{b})} = \frac{T(\frac{n}{b^2})}{g(\frac{n}{b^2})} + h(\frac{n}{b})$$

$$\frac{T(\frac{n}{b^2})}{g(\frac{n}{b^2})} = \frac{T(\frac{n}{b^3})}{g(\frac{n}{b^3})} + h(\frac{n}{b^2})$$

$$\vdots$$

$$\frac{T(n)}{g(n)} = \boxed{h(n) + h(\frac{n}{b}) + h(\frac{n}{b^2}) + \cdots} + \Theta(1)$$

Two challenges: (1) Figure out what to divide in both sides + (2) Compute the sum

Q1. Telescoping

Use telescoping to give a tight asymptotic bound for

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

A. $T(n) = \Theta(n)$
B. $T(n) = \Theta(n \log n)$
C. $T(n) = \Theta(n \log \log n)$
D. $T(n) = \Theta(n \log^2 n)$
E. $T(n) = \Theta(n^2)$

Q1. Telescoping

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

$$\frac{T(n)}{n} = \frac{T\left(\frac{n}{4}\right)}{\frac{n}{4}} + \frac{1}{\log n}$$

◀ divide both sides by n

$$\frac{T\left(\frac{n}{4}\right)}{\frac{n}{4}} = \frac{T\left(\frac{n}{16}\right)}{\frac{n}{16}} + \frac{1}{\log \frac{n}{4}}$$

$$\frac{T\left(\frac{n}{16}\right)}{\frac{n}{16}} = \frac{T\left(\frac{n}{64}\right)}{\frac{n}{64}} + \frac{1}{\log \frac{n}{16}}$$

$$\vdots$$

$$\frac{T(n)}{n} = \frac{1}{\log n} + \frac{1}{\log \frac{n}{4}} + \frac{1}{\log \frac{n}{16}} + \cdots$$

Q1. Telescoping

$$\frac{T(n)}{n} = \frac{1}{\log n} + \frac{1}{\log \frac{n}{4}} + \frac{1}{\log \frac{n}{16}} + \cdots$$

$$= \Theta\left(\frac{1}{\log n} + \frac{1}{\log n - 1} + \frac{1}{\log n - 2} + \cdots\right)$$ ◀ take log base 4

$$= \Theta(\log \log n)$$ ◀ harmonic series

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

Asymptotic Analysis Useful Facts Refresher:
$\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{n} = \ln n + O(1)$ (harmonic series)

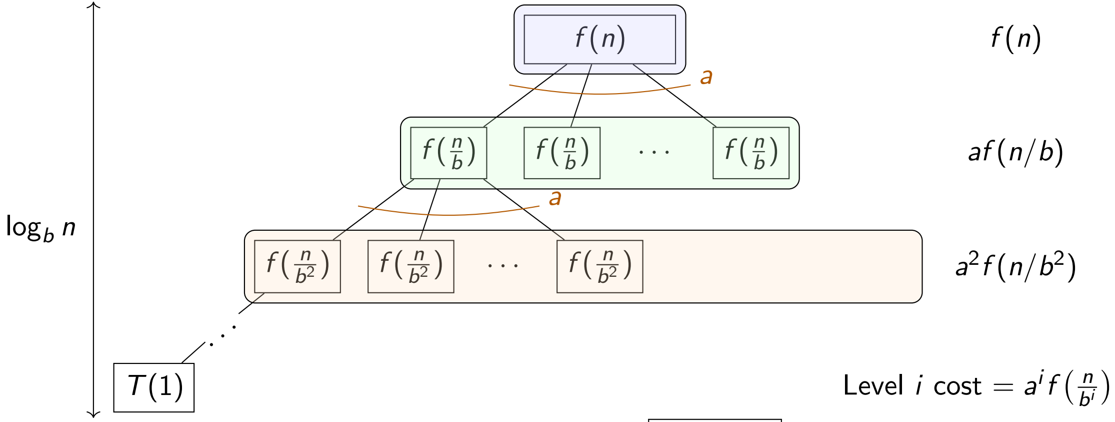
Recap: Master Theorem

Given $T(n) = aT(\frac{n}{b}) + f(n)$. There are 3 cases:

	Case	**Results**
1	$f(n) = O(n^{\log_b a - \varepsilon})$, $\varepsilon > 0$	$T(n) = \Theta(n^{\log_b a})$
2	$f(n) = \Theta(n^{\log_b a} \log^k n)$, $k \geq 0$	$T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$
3	$f(n) = \Omega(n^{\log_b a + \varepsilon})$, $\varepsilon > 0$ **Regularity condition**: $af(\frac{n}{b}) \leq kf(n)$, $k < 1$	$T(n) = \Theta(f(n))$

Recap: Master Theorem

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

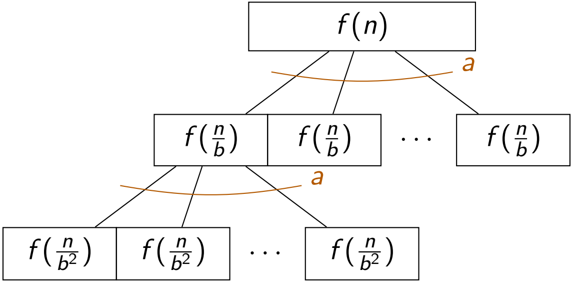


Number of nodes $= \underbrace{1 + a + a^2 + \cdots}_{\log_b n \text{ terms}} = \Theta(a^{\log_b n}) = \boxed{\Theta(n^{\log_b a})}$.

Recap: Master Theorem

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

- **Case 1** $(f(n) = O(n^{\log_b a - \varepsilon}),\ \varepsilon > 0)$: $T(n) = \Theta(n^{\log_b a})$



Recap: Master Theorem

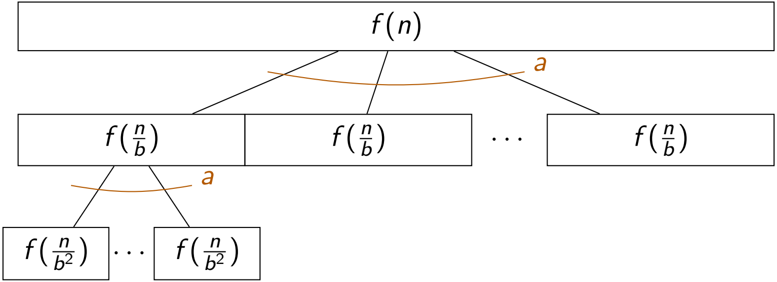
⚠ **FAQ: Why don't we use $o(n^{\log_b a})$ instead of $O(n^{\log_b a - \varepsilon})$?**

They have different meanings. For example, $f(n) = \frac{n^3}{\log n}$ is in $o(n^3)$ but **NOT** in $O(n^{3-\varepsilon})$ for all $\varepsilon > 0$.

Recap: Master Theorem

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

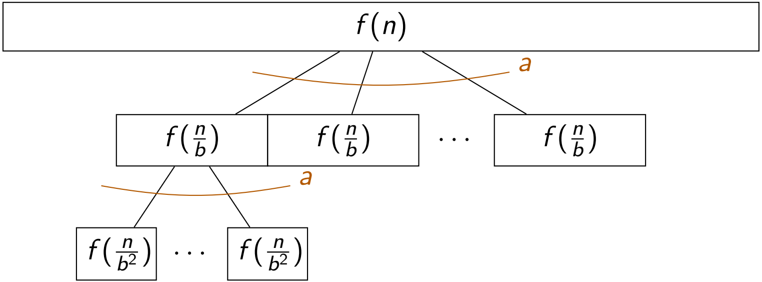
- **Case 2** ($f(n) = \Theta(n^{\log_b a} \log^k n)$, $k \geq 0$): $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$



Recap: Master Theorem

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

- **Case 3** $(f(n) = \Omega(n^{\log_b a + \varepsilon}), \varepsilon > 0$ **AND** $af(\frac{n}{b}) \leq kf(n), k < 1)$:
  $T(n) = \Theta(f(n))$



Q2. Master Theorem?

Give a tight asymptotic bound for $T(n) = 5T\left(\frac{n}{3}\right) + n$.

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

B. $T(n) = \Theta(n^{\log_5 3})$

C. $T(n) = \Theta(n^{\log_3 5})$

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

E. $T(n) = \Theta(n)$

Given $T(n) = af(\frac{n}{b}) + f(n)$.

1. $f(n) = O(n^{\log_b a - \varepsilon})$, $\epsilon > 0$: $T(n) = \Theta(n^{\log_b a})$
2. $f(n) = \Theta(n^{\log_b a} \log^k n)$, $k \geq 0$: $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$
3. $f(n) = \Omega(n^{\log_b a + \varepsilon})$, $\epsilon > 0$ **and** $af(\frac{n}{b}) \leq kf(n)$, $k < 1$: $T(n) = \Theta(f(n))$

Q2. Master Theorem Case 1

$$T(n) = 5T\left(\frac{n}{3}\right) + n$$

- Apply Master Theorem with $a = 5$, $b = 3$, $f(n) = n$.
- Let $k = \log_3 5$. $f(n) = n^1 = O(n^{k-\varepsilon})$. $\Rightarrow$ Case 1 applies.
- $T(n) = \Theta(n^{\log_3 5})$.

Given $T(n) = af(\frac{n}{b}) + f(n)$.

1. $f(n) = O(n^{\log_b a - \varepsilon})$, $\epsilon > 0$: $T(n) = \Theta(n^{\log_b a})$
2. $f(n) = \Theta(n^{\log_b a} \log^k n)$, $k \geq 0$: $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$
3. $f(n) = \Omega(n^{\log_b a + \varepsilon})$, $\epsilon > 0$ **and** $af(\frac{n}{b}) \leq kf(n)$, $k < 1$: $T(n) = \Theta(f(n))$

Q3. Master Theorem?

Give a tight asymptotic bound for $T(n) = 9T\left(\frac{n}{3}\right) + n^3$.

A. $T(n) = \Theta(n^9)$
B. $T(n) = \Theta(n^3 \log n)$
C. $T(n) = \Theta(n^2)$
D. $T(n) = \Theta(n^3)$
E. $T(n) = \Theta(n \log^2 n)$

Given $T(n) = af(\frac{n}{b}) + f(n)$.

1. $f(n) = O(n^{\log_b a - \varepsilon})$, $\epsilon > 0$: $T(n) = \Theta(n^{\log_b a})$
2. $f(n) = \Theta(n^{\log_b a} \log^k n)$, $k \geq 0$: $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$
3. $f(n) = \Omega(n^{\log_b a + \varepsilon})$, $\epsilon > 0$ **and** $af(\frac{n}{b}) \leq kf(n)$, $k < 1$: $T(n) = \Theta(f(n))$

Q3. Master Theorem Case 3

$$T(n) = 9T\left(\frac{n}{3}\right) + n^3$$

- Apply Master Theorem with $a = 9$, $b = 3$, $f(n) = n^3$.
- Let $k = \log_3 9 = 2$. $f(n) = n^3 = \Omega(n^{k+\varepsilon})$.
- $af(\frac{n}{b}) = 9 \cdot (\frac{n}{3})^3 = \frac{n^3}{3} \leq \frac{1}{3}f(n)$, so regularity condition holds. $\Rightarrow$ Case 3 applies.
- $T(n) = \Theta(n^3)$.

Given $T(n) = af(\frac{n}{b}) + f(n)$.

1. $f(n) = O(n^{\log_b a - \varepsilon})$, $\epsilon > 0$: $T(n) = \Theta(n^{\log_b a})$
2. $f(n) = \Theta(n^{\log_b a} \log^k n)$, $k \geq 0$: $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$
3. $f(n) = \Omega(n^{\log_b a + \varepsilon})$, $\epsilon > 0$ **and** $af(\frac{n}{b}) \leq kf(n)$, $k < 1$: $T(n) = \Theta(f(n))$

Q4. Master Theorem?

Give a tight asymptotic bound for $T(n) = 16T\left(\frac{n}{4}\right) + n^2 \log n$.

A. $T(n) = \Theta(n^2 \log n)$

B. $T(n) = \Theta(n^2 \log^2 n)$

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

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

E. $T(n) = \Theta(n^4 \log n)$

Given $T(n) = af(\frac{n}{b}) + f(n)$.

1. $f(n) = O(n^{\log_b a - \varepsilon})$, $\epsilon > 0$: $T(n) = \Theta(n^{\log_b a})$
2. $f(n) = \Theta(n^{\log_b a} \log^k n)$, $k \geq 0$: $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$
3. $f(n) = \Omega(n^{\log_b a + \varepsilon})$, $\epsilon > 0$ **and** $af(\frac{n}{b}) \leq kf(n)$, $k < 1$: $T(n) = \Theta(f(n))$

Q4. Master Theorem Case 2

$$T(n) = 16T\left(\frac{n}{4}\right) + n^2 \log n$$

- Apply Master Theorem with $a = 16,\ b = 4,\ f(n) = n^2 \log n$.
- Let $k = log_4 16 = 2$. Then $f(n) = \Theta(n^k \log n)$. $\Rightarrow$ Case 2 applies.
- $T(n) = \Theta(n^2 \log^2 n)$.

Given $T(n) = af(\frac{n}{b}) + f(n)$.

1. $f(n) = O(n^{\log_b a - \varepsilon}),\ \epsilon > 0$: $T(n) = \Theta(n^{\log_b a})$
2. $f(n) = \Theta(n^{\log_b a} \log^k n),\ k \geq 0$: $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$
3. $f(n) = \Omega(n^{\log_b a + \varepsilon}),\ \epsilon > 0$ **and** $af(\frac{n}{b}) \leq kf(n),\ k < 1$: $T(n) = \Theta(f(n))$

Recap: Substitution Method

- **Key idea**. Start with a guess and attempt to prove it with induction.
- Example: $T(n) = 4T(\frac{n}{2}) + n$. You guess that $T(n) = \Theta(n^2)$ and you want to prove it.
  - **Upper bound**: Show that $T(n) = O(n^2)$.
  - **Lower bound**: Show that $T(n) = \Omega(n^2)$.

Upper bound: Failed Attempt.
Show that $T(n) \leq cn^2$ for some constant c.
Induction Step:

$$\begin{aligned} T(n) &= 4T(\frac{n}{2}) + n \\ &\leq 4c(\frac{n}{2})^2 + n && \blacktriangleleft \text{ by inductive hypothesis} \\ &= cn^2 \boxed{+n} && \blacktriangleleft \text{ this is not } \leq cn^2! \end{aligned}$$

Upper bound: Fixed Attempt.
Show that $T(n) \leq c_2 n^2 - c_1 n$ for some c_1, c_2.
Induction Step:

$$\begin{aligned} T(n) &= 4T(\frac{n}{2}) + n \\ &\leq 4c_2(\frac{n}{2})^2 - 4c_1(\frac{n}{2}) + n && \blacktriangleleft \text{ by inductive hy.} \\ &= c_2 n^2 \boxed{-2c_1 n + n} \\ &\leq c_2 n^2 - c_1 n && \blacktriangleleft \text{ pick } c_1 \geq 1 \end{aligned}$$

Recap: Substitution Method

- **Key idea**. Start with a guess and attempt to prove it with induction.
- Example: $T(n) = 4T(\frac{n}{2}) + n$. You guess that $T(n) = \Theta(n^2)$ and you want to prove it.
  - **Upper bound**: Show that $T(n) = O(n^2)$.
  - **Lower bound**: Show that $T(n) = \Omega(n^2)$.

Lower bound:.
Show that $T(n) \geq cn^2$ for some constant c.
Induction Step:

$$\begin{aligned} T(n) &= 4T(\frac{n}{2}) + n \\ &\geq 4c(\frac{n}{2})^2 + n && \blacktriangleleft \text{ by inductive hypothesis} \\ &= cn^2 + n \\ &\geq cn^2 && \blacktriangleleft \text{ all good now!} \end{aligned}$$

Recap: Substitution Method

How to check if your guess is good?

- Consider $T(n) =$ **Recursive calls** $+$ **"Merging" time**
- Substitute your guess into the recurrence and check:
  1. Time contributed by **recursive calls** (Constants matter here).
  2. Time contributed by **merging** (Constants don't matter).
- Compare both against your guess:
  - If either one is larger than your guess (including by a constant factor for recursive calls), try **larger** guesses.
  - If both are smaller than your guess, try **smaller** guesses.
  - If one of them is equal to your guess, and the other is smaller than your guess, then your guess is **just right**!
  - If both of them are equal to your guess, **add a** $\log n$ **factor** (Case 2 of Master Theorem, to account for recursive calls).

Substitution Method Guessing — Example 1

Examples:

- $T(n) = \underbrace{2T(n/2) + 8T(n/4)}_{\text{recursive calls}} + \underbrace{3n^2}_{\text{"Merging"}}$.

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

- Substitute your guess
  1. Recursive calls (constants matter):

$$2\left(\frac{n}{2}\right)^3 + 8\left(\frac{n}{4}\right)^3 = \frac{3}{8}n^3 \quad (< n^3)$$

  2. Merging (constants don't matter):
     $3n^2 \quad (= o(n^3))$
- Since both terms are smaller than our guess, our guess was **too large**

Refined Guess: $T(n) = \Theta(n^2)$

- Substitute your guess
  1. Recursive calls:

$$2(\frac{n}{2})^2 + 8(\frac{n}{4})^2 = n^2 \quad (= n^2)$$

  2. Merging: $3n^2 \quad (= \Theta(n^2))$
- Since both terms are equal to our guess, we should **add a log factor** and prove $T(n) = \Theta(n^2 \log n)$

Substitution Method Guessing — Example 2

Examples:

- $T(n) = \underbrace{2T(n/2) + 8T(n/4)}_{\text{recursive calls}} + \underbrace{3n}_{\text{"Merging"}}$.

Guess: $T(n) = \Theta(n)$

- Substitute your guess
  1. Recursive calls (constants matter):
     $$2\left(\frac{n}{2}\right) + 8\left(\frac{n}{4}\right) = 3n \quad (> n)$$
  2. Merging (constants don't matter): $3n \quad (= \Theta(n))$
- Since the recursive calls are larger than our guess, our guess was **too small**

Refined Guess: $T(n) = \Theta(n^2)$

- Substitute your guess
  1. Recursive calls:
     $$2\left(\frac{n}{2}\right)^2 + 8\left(\frac{n}{4}\right)^2 = n^2 \quad (= n^2)$$
  2. Merging: $3n \quad (= o(n^2))$
- Since one term equals our guess and the other is smaller, our guess is **just right** and we should prove $T(n) = \Theta(n^2)$

Q5. Substitution Method

Give a tight asymptotic bound for $T(n) = 4T\left(\frac{n}{2}\right) + \sqrt{n}$ using the substitution method.

Q5. Substitution Method

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

Guess: $T(n) = \Theta(n^2)$.

- (Upper bound) Want to show: $T(n) \leq c_2 n^2 - c_1 n$
- (Lower bound) Want to show: $T(n) \geq cn^2$

Question 5B: What is a good choice of c_1, c_2 and c?

Q5. Substitution Method

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

Guess: $T(n) = \Theta(n^2)$.
(Upper bound) Want to show: $T(n) \leq c_2 n^2 - c_1 n$ for $c_1 = 1$, $c_2 = T(1) + c_1$.

Basis Step: $T(1) = c_2 - c_1$.
Induction Step:

$$\begin{aligned}
T(n) &= 4T\left(\frac{n}{2}\right) + \sqrt{n} \\
&\leq 4\left(c_2\left(\frac{n}{2}\right)^2 - c_1\left(\frac{n}{2}\right)\right) + \sqrt{n} && \blacktriangleleft \text{ by induction hypothesis} \\
&= c_2 n^2 - 2c_1 n + \sqrt{n} \\
&= c_2 n^2 - c_1 n - (c_1 n - \sqrt{n}) \\
&\leq c_2 n^2 - c_1 n && \blacktriangleleft \text{ Note that } n - \sqrt{n} \geq 0
\end{aligned}$$

Q5. Substitution Method

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

Guess: $T(n) = \Theta(n^2)$.
(Lower bound) Want to show: $T(n) \geq cn^2$ for $c = T(1)$.

Basis Step: $T(1) = c$.
Induction Step:

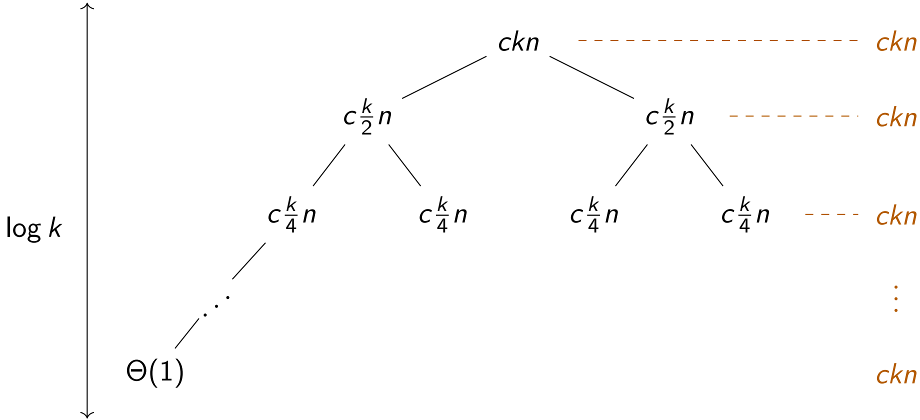
$$\begin{aligned} T(n) &= 4T\left(\frac{n}{2}\right) + \sqrt{n} \\ &\geq 4\left(c\left(\frac{n}{2}\right)^2\right) + \sqrt{n} && \blacktriangleleft \text{ by induction hypothesis} \\ &= cn^2 + \sqrt{n} \\ &\geq cn^2 \end{aligned}$$

Q6. Merging Sorted Arrays

- Suppose you are given k sorted arrays, $A_1, A_2, \ldots, A_k$, with n elements each, and you are to merge them.
- Let $T(k, n)$ denote the complexity of merging k arrays of size n.
- Suppose you decide the best way to do above is recursion (when $k > 1$)
  1. Merge the first $\lceil \frac{k}{2} \rceil$ arrays of size n
  2. Merge the remaining $\lfloor \frac{k}{2} \rfloor$ arrays of size n
  3. Merge the two arrays obtained in the first two steps above
- Give a formula for $T(k, n)$ based on the above. Solve your recursion formula.

Q6. Merging Sorted Arrays

$$T(k, n) = 2T\left(\frac{k}{2}, n\right) + ckn \text{ (for some constant } c\text{)}$$



Total cost: $\Theta(kn \log k)$.

Extended Master Theorem

Extra Slide

Disclaimer: You CANNOT quote this directly in assignments / exams.

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

- **Case 2**: $f(n) = \Theta(n^{\log_b a} \log^k n)$, $k \geq 0$. What if $k < 0$?

Extended Master Theorem:

- If $k < -1$, then $T(n) = \Theta(n^{\log_b a})$.
- If $k = -1$, then $T(n) = \Theta(n^{\log_b a} \log \log n)$.
- If $k > -1$, then $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$.

Extended Master Theorem

Extra Slide

$$T(n) = aT(\frac{n}{b}) + n^{\log_b a} \log^k n$$

$$\frac{T(n)}{n^{\log_b a}} = \frac{T(\frac{n}{b})}{(\frac{n}{b})^{\log_b a}} + \log^k n$$

◀ Divide $n^{\log_b a}$ on both sides

$$\frac{T(\frac{n}{b})}{(\frac{n}{b})^{\log_b a}} = \frac{T(\frac{n}{b^2})}{(\frac{n}{b^2})^{\log_b a}} + \log^k(\frac{n}{b})$$

$$\frac{T(\frac{n}{b^2})}{(\frac{n}{b^2})^{\log_b a}} = \frac{T(\frac{n}{b^3})}{(\frac{n}{b^3})^{\log_b a}} + \log^k(\frac{n}{b^2})$$

$$\frac{T(\frac{n}{b^3})}{(\frac{n}{b^3})^{\log_b a}} = \frac{T(\frac{n}{b^4})}{(\frac{n}{b^4})^{\log_b a}} + \log^k(\frac{n}{b^3})$$

$$\vdots$$

$$\frac{T(n)}{n^{\log_b a}} = \log^k n + \log^k(\frac{n}{b}) + \log^k(\frac{n}{b^2}) + \cdots + \Theta(1)$$

◀ Telescoping series

Extended Master Theorem

Extra Slide

Assuming logarithms are in base b,

$$\frac{T(n)}{n^{\log_b a}} = \log^k n + \log^k(\frac{n}{b}) + \log^k(\frac{n}{b^2}) + \cdots + \Theta(1)$$
$$= (\log n)^k + (\log n - 1)^k + (\log n - 2)^k + \cdots + \Theta(1)$$

- If $k < -1$,

$$\sum_{x=1}^{\log n} x^k = \Theta(1) \blacktriangleleft \text{ } p\text{-series (Check MA1521/MA2002 Notes)}$$

- If $k = -1$,

$$\frac{1}{\log n} + \frac{1}{\log n - 1} + \frac{1}{\log n - 2} + \cdots = \Theta(\log(\log n)) \blacktriangleleft \text{ harmonic series}$$

Q1 (again). Extended Master Theorem

Extra Slide

Use ~~telescoping~~ Extended Master Theorem to give a tight asymptotic bound for

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

- Apply extended master theorem with $f(n) = n \log^{-1} n$.
- $T(n) = \Theta(n \log \log n)$.

Given $T(n) = aT(\frac{n}{b}) + f(n)$, where $f(n) = \Theta(n^{\log_b a} \log^k n)$.
2a. If $k < -1$, then $T(n) = \Theta(n^{\log_b a})$.
2b. If $k = -1$, then $T(n) = \Theta(n^{\log_b a} \log \log n)$.
2c. If $k > -1$, then $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$.

Extended Master Theorem

Exercises: (try them out at home using telescoping series)

1. (AY 19/20 Sem 2 Midterm; $k = -1$) $T(n) = 9T(n/3) + n^2/\log n$
2. (AY 20/21 Sem 2 Midterm; $k < -1$) $T(n) = 27T(n/3) + n^3/\log^2 n$

Bonus. More Recurrences

Extra Slide

Note: Bonus questions are entirely optional. They can get very challenging and/or fall outside the course curriculum. Interested individuals may attempt.

Solve the following recurrences (using Master Theorem indirectly):

1. $T(n) = T(n/3) + T(n/6) + 5n^{\sqrt{\log n}}$.
2. $T(n) = 4T(\sqrt{n}) + 6\log^2 n$.

Bonus. More Recurrences

Extra Slide

1. $T(n) = T(n/3) + T(n/6) + 5n^{\sqrt{\log n}}$.

Solution:

- Note that $T(n) \leq 2T(n/3) + 5n^{\sqrt{\log n}}$.
  - Apply Master Theorem with $a = 2$, $b = 3$, $f(n) = 5n^{\sqrt{\log n}}$.
  - $f(n) = \Omega(n^{\log_3 2})$ and $2T(n/3) = 10\left(\frac{n}{3}\right)^{\sqrt{\log \frac{n}{3}}} < 10 \cdot (\frac{1}{3})^{\sqrt{\log n/3}} \cdot n^{\sqrt{\log n}}$ $< \frac{2}{3} \cdot 5n^{\sqrt{\log n}}$ (when $n > 8$). $\Rightarrow$ Case 3 applies.
  - Therefore, $T(n) = O(5n^{\sqrt{\log n}})$.
- It is also easy to see that $T(n) \geq 5n^{\sqrt{\log n}}$. Therefore, $T(n) = \Omega(5n^{\sqrt{\log n}})$.
- Hence $T(n) = \Theta(n^{\sqrt{\log n}})$.

Bonus. More Recurrences

2. $T(n) = 4T(\sqrt{n}) + 6\log^2 n$.

Solution:

- Substitute $n = 2^m$, we have $T(2^m) = 4T(2^{m/2}) + 6m^2$.
- Next, substitute $S(m) = T(2^m)$, we have $S(m) = 4S(m/2) + 6m^2$.
- Apply Master Theorem with $a = 4$, $b = 2$ and $f(n) = 6m^2$.
- $f(n) = \Theta(n^{\log_b a}) \Rightarrow$ Case 2 applies.
- Therefore, $S(m) = \Theta(m^2 \log m)$.
- Substituting back, we have $T(2^m) = \Theta(m^2 \log m)$, i.e. $T(n) = \Theta((\log n)^2 \log\log n)$.