Master’s method

Asymptotic Behavior of Recursive Algorithms

Asymptotic Behavior of Recursive Algorithms

---master theorem

- When analyzing algorithms, recall that we only care about the asymptotic behavior
- Recursive algorithms are no different
- Rather than solving exactly the recurrence relation associated with the cost of an algorithm, it is sufficient to give an asymptotic characterization
- The main tool for doing this is the master theorem

Master Theorem

Given a recurrence of the form

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

T(n) be a monotonically increasing function that satisfies

for constants $a\ (\geq 1)$) and $b\ (> 1)$ with f asymptotically positive, the following statements are true:

- **Case 1.** If $f(n) = O\left(n^{\log_b a - \epsilon}\right)$ for some $\epsilon > 0$, then $T(n) = \Theta\left(n^{\log_b a}\right)$.
- **Case 2.** If $f(n) = \Theta\left(n^{\log_b a}\right)$, then $T(n) = \Theta\left(n^{\log_b a} \log n\right)$.
- **Case 3.** If $f(n) = \Omega\left(n^{\log_b a + \epsilon}\right)$ for some $\epsilon > 0$ $\left(\text{and } af\left(\frac{n}{b}\right) \leq cf(n) \text{ for some } c < 1 \text{ for all } n \text{ sufficiently large}\right)$, then $T(n) = \Theta\big(f(n)\big)$.

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

$f(n)$ is asymptotically smaller or larger than $n^{\log_b a}$ by a polynomial factor n^{ε}

$f(n)$ is asymptotically equal with $n^{\log_b a}$

Master Theorem (Simple)

- Let T(n) be a monotonically increasing function that satisfies

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

$$T(1) = c$$

where $a \geq 1$, $b \geq 2$, $c>0$. If f(n) is $\Theta(n^d)$ where $d \geq 0$ then

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{If } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

$f(n)=n^{\log_b(a)}$

d > log(a) [base b] // $n^{\log_b a-e}$

d = log(a) [base b] // $n^{\log_b a}$

d < log(a) [base b] // $n^{\log_b a +e}$

Proof: https://www.cs.cornell.edu/courses/cs3110/2012sp/lectures/lec20-master/mm-proof.pdf

Master Theorem: Pitfalls

- You cannot use the Master Theorem if
  - $T(n)$ is not monotone, e.g. $T(n) = \sin(x)$
  - $f(n)$ is not a polynomial

- Note that the Master Theorem does not solve all the recurrence equation

Master Theorem: Example 1

- Let $T(n) = T(n/2) + ½\, n^2 + n$. What are the parameters?

  a = 1

  b = 2

  d = 2

  Therefore, which condition applies?

  $1 < 2^2$, case 1 applies

- We conclude that

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

- Let T(n) be <u>a monotonically increasing</u> function that satisfies

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

  $T(1) = c$

  where $a \geq 1$, $b \geq 2$, $c>0$. If $f(n)$ is $\Theta(n^d)$ where $d \geq 0$ then

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{If } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

Master Theorem: Example 2

- Let T(n)= 2 T(n/4) + √n + 42. What are the parameters?

  a = 2

  b = 4

  d = 1/2

  Therefore, which condition applies?

  $2 = 4^{1/2}$, case 2 applies

- We conclude that

$$T(n) \in \Theta(n^d \log n) = \Theta(\log n \sqrt{n})$$

- Let T(n) be <u>a monotonically increasing</u> function that satisfies

  T(n) = a T(n/b) + f(n)

  T(1) = c

  where $a \geq 1$, $b \geq 2$, $c>0$. If f(n) is $\Theta(n^d)$ where $d \geq 0$ then

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{If } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

Master Theorem: Example 3

- Let T(n)= 3 T(n/2) + 3/4n + 1. What are the parameters?

a = 3

b = 2

d = 1

Therefore, which condition applies?

$3 > 2^1$, case 3 applies

- Let T(n) be <u>a monotonically increasing</u> function that satisfies

  T(n) = a T(n/b) + f(n)

  T(1) = c

  where $a \geq 1$, $b \geq 2$, c>0. If f(n) is $\Theta(n^d)$ where $d \geq 0$ then

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{If } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

- We conclude that

$$T(n) \in \Theta(n^{\log_b a}) = \Theta(n^{\log_2 3})$$

- Note that $\log_2 3 \approx 1.584\ldots$, can we say that $T(n) \in \Theta(n^{1.584})$

Examples

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

$a = 2, b = 2,$

$\Rightarrow f(n) = \Theta(n) \Rightarrow$ Case 2

$\Rightarrow T(n) = \Theta(n\lg n)$

- Let T(n) be a monotonically increasing function that satisfies

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

  $T(1) = c$

  where $a \geq 1$, $b \geq 2$, $c>0$. If $f(n)$ is $\Theta(n^d)$ where $d \geq 0$ then

$$T(n) = \begin{cases} \Theta(n^d) & \text{if } a < b^d \\ \Theta(n^d \log n) & \text{If } a = b^d \\ \Theta(n^{\log_b a}) & \text{if } a > b^d \end{cases}$$

Examples

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

$a = 2, b = 2,$

Compare n with $f(n) = n^2$

$\Rightarrow f(n) = \Omega(n^{1+\varepsilon})$ Case 3 $\Rightarrow$ verify regularity cond.

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

$\Leftrightarrow 2\, n^2/4 \leq c\, n^2 \Rightarrow c = \frac{1}{2}$ is a solution $(c<1)$

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

Examples (cont.)

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

$a = 2$, $b = 2$, Compare n with $f(n) = n^{1/2}$

$\Rightarrow f(n) = O(n^{1-\varepsilon})$ Case 1

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

‘Fourth’ Condition

- Recall that we cannot use the Master Theorem if f(n), the non-recursive cost, is not a polynomial
- There is a limited 4th condition of the Master Theorem that allows us to consider polylogarithmic functions
- **Corollary**: If $f(n) \in \Theta(n^{\log_b a} \log^k n)$ for some k≥0 then

$$T(n) \in \Theta(n^{\log_b a} \log^{k+1} n)$$

- This final condition is fairly limited, and we present it merely for sake of completeness.. Relax ☺

‘Fourth’ Condition: Example

- Say we have the following recurrence relation

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

- Clearly, a=2, b=2, but f(n) is not a polynomial. However, we have $f(n) \in \Theta(n \log n)$, k=1
- Therefore by the 4th condition of the Master Theorem we can say that

$$T(n) \in \Theta(n^{\log_b a} \log^{k+1} n) = \Theta(n^{\log_2 2} \log^2 n) = \Theta(n \log^2 n)$$

Examples

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

$a = 2,\ b = 2,\ \log_2 2 = 1$

- Compare n with $f(n) = n\lg n$
  - seems like case 3 should apply
- $f(n)$ must be polynomially larger by a factor of n^{ε}
- In this case it is only larger by a factor of $\lg n$

Quick Summations – Review

Review on Summations

- **Constant Series:** For integers a and b, $a \le b$,

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

- **Linear Series (Arithmetic Series):** For $n \ge 0$,

$$\sum_{i=1}^{n} i = 1 + 2 + \cdots + n = \frac{n(n+1)}{2}$$

- **Quadratic Series:** For $n \ge 0$,

$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \cdots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

Not all problem are solvable in particular time
Some problem are difficult to solve and find the solution too…