Master Method/Theorem

- Given: a *divide and conquer* algorithm
  - An algorithm that divides the problem of size n into a subproblems, each of size n/b
  - Let the cost of each stage (i.e., the work to divide the problem + combine solved subproblems) be described by the function f(n)
- Then, the Master Theorem gives us a cookbook for the algorithm's running time:

Master Method/Theorem

- Theorem 4.1 (page 94)
  - for $T(n) = aT(n/b)+f(n)$, n/b may be $\lceil n/b \rceil$ or $\lfloor n/b \rfloor$.
  - where $a \geq 1$, $b>1$ are positive integers, $f(n)$ be a non-negative function.

1. If $f(n)=O(n^{\log_b a-\varepsilon})$ for some $\varepsilon>0$, then $T(n)= \Theta(n^{\log_b a})$.
2. If $f(n)= \Theta(n^{\log_b a})$, then $T(n)= \Theta(n^{\log_b a} \lg n)$.
3. If $f(n)=\Omega(n^{\log_b a+\varepsilon})$ for some $\varepsilon>0$, and if $af(n/b) \leq cf(n)$ for some $c<1$ and all sufficiently large n, then $T(n)= \Theta(f(n))$.

The Master Theorem

- if T(n) = aT(n/b) + f(n) then

$$T(n) = \begin{cases} \Theta\left(n^{\log_b a}\right) & f(n) = O\left(n^{\log_b a - \varepsilon}\right) \\ \Theta\left(n^{\log_b a} \log n\right) & f(n) = \Theta\left(n^{\log_b a}\right) \\ \Theta\left(f(n)\right) & f(n) = \Omega\left(n^{\log_b a + \varepsilon}\right) \text{AND} \\ & af(n/b) < cf(n) \text{ for large } n \end{cases} \quad \begin{matrix} \varepsilon > 0 \\ c < 1 \end{matrix}$$

Implications of Master Theorem

- Comparison between $f(n)$ and $n^{\log_b a}$ (<,=,>)
- Must be asymptotically smaller (or larger) by a polynomial, i.e., n^{ε} for some $\varepsilon>0$.
- In case 3, the "regularity" must be satisfied, i.e., $af(n/b) \leq cf(n)$ for some $c<1$.
- There are gaps
  - between 1 and 2: $f(n)$ is smaller than $n^{\log_b a}$, but not polynomially smaller.
  - between 2 and 3: $f(n)$ is larger than $n^{\log_b a}$, but not polynomially larger.
  - in case 3, if the "regularity" fails to hold.

Application of Master Theorem

- $T(n) = 9T(n/3)+n$;
  - a=9,b=3, $f(n)$ =n
  - $n^{\log_b a} = n^{\log_3 9} = \Theta\ (n^2)$
  - $f(n)=O(n^{\log_3 9-\varepsilon})$ for ε=1
  - By case 1, $T(n) =\Theta\ (n^2)$.
- $T(n) = T(2n/3)+1$
  - a=1,b=3/2, $f(n)$ =1
  - $n^{\log_b a} = n^{\log_{3/2} 1} = \Theta\ (n^0) = \Theta\ (1)$
  - By case 2, $T(n)= \Theta(\lg n)$.

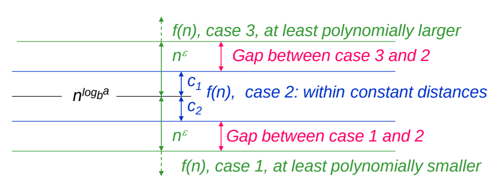
Application of Master Theorem

- $T(n) = 3T(n/4)+n\lg n$;
  - a=3,b=4, $f(n)$ =$n\lg n$
  - $n^{\log_b a} = n^{\log_4 3} = \Theta\ (n^{0.793})$
  - $f(n)= \Omega(n^{\log_4 3+\varepsilon})$ for ε≈0.2
  - Moreover, for large n, the "regularity" holds for c=3/4.
    - o $af(n/b) =3(n/4)\lg (n/4) \le (3/4)n\lg n = cf(n)$
  - By case 3, $T(n) =\Theta\ (f(n))=\Theta\ (n\lg n)$.

Exception to Master Theorem

- $T(n) = 2T(n/2)+n\lg n$;
  - a=2,b=2, $f(n)$ =$n\lg n$
  - $n^{\log_b a} = n^{\log_2 2} = \Theta\ (n)$
  - $f(n)$ is asymptotically larger than $n^{\log_b a}$, but not polynomially larger because
  - $f(n)/n^{\log_b a} = \lg n$, which is asymptotically less than n^{ε} for any ε>0.
  - Therefore,this is a gap between 2 and 3.

Where Are the Gaps

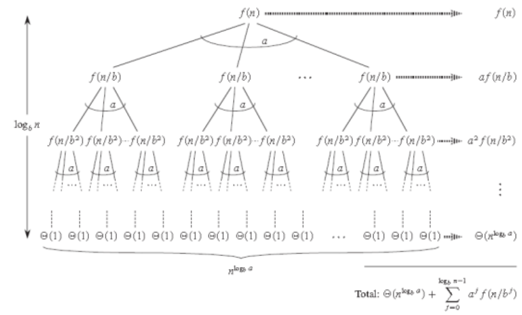


Note: 1. for case 3, the regularity also must hold.
2. if f(n) is lg n smaller, then fall in gap in 1 and 2
3. if f(n) is lg n larger, then fall in gap in 3 and 2
4. if $f(n)=\Theta(n^{\log_b a}\lg^k n)$, then $T(n)=\Theta(n^{\log_b a}\lg^{k+1} n)$. (as exercise)

Proof of Master Theorem

- The proof for the exact powers, $n=b^k$ for $k\ge1$.
- Lemma 4.2
  - for $T(n) = \begin{cases}\Theta(1) & \text{if } n=1\\ aT(n/b)+f(n) & \text{if } n=b^k \text{ for } k\ge1\end{cases}$
  - where $a \ge 1$, $b>1$, $f(n)$ be a nonnegative function,
  - Then
  - $T(n) = \Theta(n^{\log_b a})+\sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$
- Proof:
  - By iterating the recurrence
  - By recursion tree (See figure 4.7)

Recursion tree for $T(n)=aT(n/b)+f(n)$



Proof of Master Theorem (cont.)

- Lemma 4.3:
  - Let $a \ge 1$, $b>1$, $f(n)$ be a nonnegative function defined on exact power of b, then
  - $g(n)=\sum_{j=0}^{\log_b n-1} a^j f(n/b^j)$ can be bounded for exact power of b as:

1. If $f(n)=O(n^{\log_b a-\varepsilon})$ for some $\varepsilon>0$, then $g(n)= O(n^{\log_b a})$.
2. If $f(n)= \Theta(n^{\log_b a})$, then $g(n)= \Theta(n^{\log_b a} \lg n)$.
3. If $f(n)= \Omega(n^{\log_b a+\varepsilon})$ for some $\varepsilon>0$ and if $af(n/b) \le cf(n)$ for some $c<1$ and all sufficiently large $n \ge b$, then $g(n)= \Theta(f(n))$.

Proof of Lemma 4.3

- For case 1: $f(n)=O(n^{\log_b a-\varepsilon})$ implies $f(n/b^j)=O((n/b^j)^{\log_b a-\varepsilon})$, so
- $g(n)=\sum_{j=0}^{\log_b n-1} a^j f(n/b^j) = O(\sum_{j=0}^{\log_b n-1} a^j (n/b^j)^{\log_b a-\varepsilon})$

$$= O(n^{\log_b a-\varepsilon}\sum_{j=0}^{\log_b n-1} a^j/(b^{\log_b a-\varepsilon})^j) = O(n^{\log_b a-\varepsilon}\sum_{j=0}^{\log_b n-1} a^j/(a^j(b^{-\varepsilon})^j))$$

$$= O(n^{\log_b a-\varepsilon}\sum_{j=0}^{\log_b n-1} (b^{\varepsilon})^j) = O(n^{\log_b a-\varepsilon}(((b^{\varepsilon})^{\log_b n}-1)/(b^{\varepsilon}-1)))$$

$$= O(n^{\log_b a-\varepsilon}(((b^{\log_b n})^{\varepsilon}-1)/(b^{\varepsilon}-1)))=O(n^{\log_b a}\, n^{-\varepsilon}(n^{\varepsilon}-1)/(b^{\varepsilon}-1))$$

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

Proof of Lemma 4.3(cont.)

- For case 2: $f(n)=\Theta(n^{\log_b a})$ implies $f(n/b^j)=\Theta((n/b^j)^{\log_b a})$, so

$$g(n)=\sum_{j=0}^{\log_b n-1} a^j f(n/b^j) = \Theta\left(\sum_{j=0}^{\log_b n-1} a^j (n/b^j)^{\log_b a}\right)$$

$$=\Theta\left(n^{\log_b a}\sum_{j=0}^{\log_b n-1} a^j/(b^{\log_b a})^j\right)=\Theta\left(n^{\log_b a}\sum_{j=0}^{\log_b n-1} 1\right)$$

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

L4.13

Proof of Lemma 4.3(cont.)

- For case 3:
  - Since $g(n)$ contains $f(n)$, $g(n)=\Omega(f(n))$
  - Since $af(n/b)\le cf(n)$, $a^j f(n/b^j)\le c^j f(n)$, why???
  - $g(n)=\sum_{j=0}^{\log_b n-1} a^j f(n/b^j) \le \sum_{j=0}^{\log_b n-1} c^j f(n) \le f(n)\sum_{j=0}^{\infty} c^j$
  - $=f(n)(1/(1-c))=O(f(n))$
  - Thus, $g(n)=\Theta(f(n))$

L4.14

Proof of Master Theorem (cont.)

- Lemma 4.4:
  - for $T(n)=\begin{cases}\Theta(1) & \text{if } n=1\\ aT(n/b)+f(n) & \text{if } n=b^k \text{ for } k\ge 1\end{cases}$
  - where $a\ge 1$, $b>1$, $f(n)$ be a nonnegative function,

1. If $f(n)=O(n^{\log_b a-\varepsilon})$ for some $\varepsilon>0$, then $T(n)=\Theta(n^{\log_b a})$.
2. If $f(n)=\Theta(n^{\log_b a})$, then $T(n)=\Theta(n^{\log_b a}\lg n)$.
3. If $f(n)=\Omega(n^{\log_b a+\varepsilon})$ for some $\varepsilon>0$, and if $af(n/b)\le cf(n)$ for some $c<1$ and all sufficiently large n, then $T(n)=\Theta(f(n))$.

L4.15

Proof of Lemma 4.4 (cont.)

- Combine Lemma 4.2 and 4.3,
  - For case 1:
    - $T(n)=\Theta(n^{\log_b a})+O(n^{\log_b a})=\Theta(n^{\log_b a})$.
  - For case 2:
    - $T(n)=\Theta(n^{\log_b a})+\Theta(n^{\log_b a}\lg n)=\Theta(n^{\log_b a}\lg n)$.
  - For case 3:
    - $T(n)=\Theta(n^{\log_b a})+\Theta(f(n))=\Theta(f(n))$ because $f(n)=\Omega(n^{\log_b a+\varepsilon})$.

L4.16

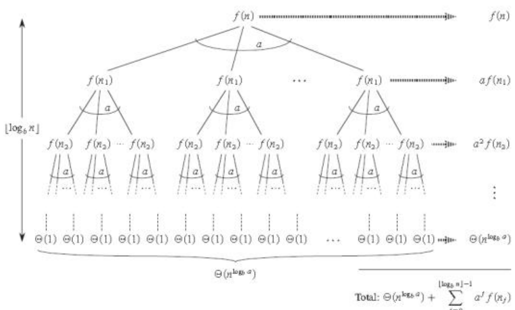
Floors and Ceilings

- $T(n)=aT(\lfloor n/b \rfloor)+f(n)$ and $T(n)=aT(\lceil n/b \rceil)+f(n)$
- Want to prove both equal to $T(n)=aT(n/b)+f(n)$
- Two results:
  - Master theorem applied to all integers n.
  - Floors and ceilings do not change the result.
    - (Note: we proved this by domain transformation too).
- Since $\lfloor n/b \rfloor \le n/b$, and $\lceil n/b \rceil \ge n/b$, upper bound for floors and lower bound for ceiling is held.
- So prove upper bound for ceilings (similar for lower bound for floors).

Upper bound of proof for $T(n)=aT(\lceil n/b \rceil)+f(n)$

- consider sequence $n, \lceil n/b \rceil, \lceil \lceil n/b \rceil /b \rceil, \lceil \lceil \lceil n/b \rceil /b \rceil /b \rceil$, …
- Let us define n_j as follows:
- $n_j = n$ if $j=0$
- $= \lceil n_{j-1}/b \rceil$ if $j>0$
- The sequence will be $n_0, n_1, \ldots, n_{\lfloor \log_b n \rfloor}$
- Draw recursion tree:

Recursion tree of $T(n)=aT(\lceil n/b \rceil)+f(n)$



The proof of upper bound for ceiling

- $T(n) = \Theta(n^{\log_b a}) + \sum_{j=0}^{\lfloor \log_b n \rfloor -1} a^j f(n_j)$
- Thus similar to Lemma 4.3 and 4.4, the upper bound is proven.

The simple format of master theorem

- $T(n)=aT(n/b)+cn^k$, with a, b, c, k are positive constants, and $a\geq 1$ and $b\geq 2$,
- $$T(n) = \begin{cases} O(n^{\log_b a}), & \text{if } a>b^k. \\ O(n^k \log n), & \text{if } a=b^k. \\ O(n^k), & \text{if } a<b^k. \end{cases}$$

Summary

Recurrences and their bounds

- Substitution
- Recursion tree
- Master theorem.
- Proof of subtleties
- Recurrences that Master theorem does not apply to.

Coding for Interview Website:
http://codingforinterviews.com/practice