Master’s Theorem

- Master theorem

General formula that works if recurrence has the form :

$$T(n) = aT(n/b) + f(n) \qquad \rightarrow (1)$$

- a is number of subproblems
- n/b is size of each subproblem
- f(n) is cost of non-recursive part

Eqn (1) describes the running time of an algorithm that divides a problem of size n into a subproblems, each of size $n=b$,where a and b are positive constants. The a subproblems are solved recursively, each in time $T(n/b)$.The function $f(n)$ encompasses the cost of dividing the problem and combining the results of the subproblems.

Consider a recurrence of the form

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

with $a>=1$, $b>1$, and $f(n)$ eventually positive.

a. If $f(n) = O(n^{\log_b(a)-\varepsilon})$, then $T(n)=\Theta(n^{\log_b(a)})$.
b. If $f(n) = \Theta(n^{\log_b(a)})$, then $T(n)=\Theta(n^{\log_b(a)}\log(n))$.
c. If $f(n) = \Omega(n^{\log_b(a)+\varepsilon})$ and $f(n)$ is regular, then $T(n)=\Theta(f(n))$

[$f(n)$ regular iff eventually $af(n/b)<= cf(n)$ for some constant $c<1$]

Examples

$$\boldsymbol{T(n) = 4T(n/2) + n}$$

$$a = 4,\ b = 2$$

$$\Rightarrow n^{\log_b a} = n^2;$$

$$f(n) = n.$$

CASE 1: $f(n) = O(n^{2-\varepsilon})$ for $\varepsilon = 1$.

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

$$\boldsymbol{T(n) = 4T(n/2) + n^2}$$

$a = 4,\ b = 2$

$\Rightarrow n^{\log_b a} = n^2;$

$f(n) = n^2.$

Case 2: $f(n) = \Theta(n^2 \lg^0 n)$, that is, $k = 0$.

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

$\boldsymbol{T(n) = 4T(n/2) + n^3}$

$a = 4, b = 2$

$\Rightarrow n^{\log_b a} = n^2;$

$f(n) = n^3.$

Case 3: $f(n) = \Omega(n^{2+\varepsilon})$ for $\varepsilon = 1$

and $4(cn/2)^3 \leq cn^3$

for $c = 1/2$.

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

$T(n) = 9T(n/3) + n$

$a=9$, $b=3$, $f(n) = n$

$n^{\log_b a} = n^{\log_3 9} = \Theta(n^2)$

Since $f(n) = O(n^{\log_3 9 - \varepsilon})$, where $\varepsilon=1$, CASE 1 applies:

Thus the solution is **$T(n) = \Theta(n^2)$**

$T(n) = T(2n/3) + 1$

$a=1$, $b=3/2$, $f(n) = 1$

$n^{\log_b a} = n^{\log_{3/2} 1}$

$f(n) = 1$

Since $f(n) = \Theta(n^{\log_b a}) = \Theta(1)$, CASE 2 applies:

Thus the solution is $\mathbf{T(n) = \Theta(lgn)}$

$T(n) = 3T(n/4) + n\lg n$

$a=3$, $b=4$, $f(n) = n\lg n$

$n^{\log_b a} = n^{\log_4 3} = n^{0.793}$

Since $f(n) = \Omega(n^{\log_4 3+\varepsilon})$ where ε is approximately 0.2

CASE 3 applies:

For sufficiently large n

$af(n/b)=3f(n/4)\lg(n/4) \le (3/4)n\lg n = cf(n)$ for $c=3/4$

By case 3

Thus the solution is **$T(n) = \Theta(n\lg n)$**

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

  $a=2,\ b=2,\ f(n) = n\lg n$

  $n^{\log_b a} = n$

  Since $f(n) = n\lg n$ is asymptotically larger than $n^{\log_b a} = n$

  CASE 3 applies:

  But $f(n) = n\lg n$ is NOT polynomially larger than $n^{\log_b a} = n$

  The ratio $f(n) / n^{\log_b a} = (n\lg n) / n$ is asymptotically less than n^{ε}

  For any positive constant ε

  So the recurrence falls the gap between case 2 and case 3

  Master theorem CANNOT be applied to the recurrence