Master’s Theorem

- Master’s Theorem is a method for solving recurrence relation of the form

$$T(n) = aT(n/b) + f(n), \; a >= 1 \text{ and } b > 1$$

- where, n = size of input
- a = number of sub-problems and n/b is the size of each sub-problem.
- F(n) is the cost of decomposition

- If $T(n) = aT(n/b) + f(n)$ where **a >= 1 and b > 1**, Master's theorem states that:

1. If $f(n) = O(n^{\log_b a-\epsilon})$, then $T(n) = \Theta(n^{\log_b a})$.
2. If $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(f(n) * \log n)$.
3. If $f(n) = \Omega(n^{\log_b a+\epsilon})$, then $T(n) = \Theta(f(n))$.

$\epsilon > 0$ is a constant.

- Solve $T(n) = 4T(n/2) + n^2$ using master's theorem

Here, $a = 4$ $b = 2$ $f(n) = n^2$

$\log_b a = \log_2 4 = 2$ ie.

$n^{\log_b a} = n^{\log_b a} = n^2$ ($n^{\log_b a}$ and $f(n)$ are same..case 2)

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

- Solve $T(n) = 8T(n/2) + n^2$ using master's theorem

Here, $a = 8$ $b = 2$ $f(n) = n^2$

$\log_b a = \log_2 8 = 3$ ie.

$n^{\log_b a} = n^{\log_b a} = n^3$ ($n^{\log_b a} > f(n)$ are same..case 1)

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

- Solve $T(n) = 3T(n/2) + n^2$ using master's theorem

Here, $a = 3$ $b = 2$ $f(n) = n^2$

$\log_b a = \log_2 3 = 1.58$ ie.

$n^{\log_b a} = n^{\log_b a} = n^{1.58}$ ($n^{\log_b a} < f(n)$..case 3)

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

- Solve $T(n) = 9T(n/3) + n^2$ using master's theorem

Here, $a = 9$ $b = 3$ $f(n) = n^2$

$\log_b a = \log_3 9 = 2$ ie.

$n^{\log_b a} = n^{\log_b a} = n^2$ ($n^{\log_b a}$ and $f(n)$ are same..case 1)

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

- Solve $T(n) = 2T(n/4) + \sqrt{n}$ using master's theorem

Here, $a = 2$ $b = 4$ $f(n) = \sqrt{n}$

$\log_b a = \log_4 2 = .5$ ie.

$n^{\log_b a} = n^{\log_b a} = n^{.5}$ ($n^{\log_b a} = f(n)$..case 2)

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

- Solve $T(n) = 2^n T(n/2) + n^n$ using master's theorem

Here, $a = 2^n$ $b = 2$ $f(n) = n^n$

$\log_b a = \log_2 2^n = n$ ie.

$n^{\log_b a} = n^{\log_b a} = n^n$ ($n^{\log_b a}$ and $f(n)$ are same..case 2)

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

- Solve $T(n) = 7T(n/2) + n^2$ using master's theorem

Here, $a = 7$ $b = 2$ $f(n) = n^2$

$\log_b a = \log_2 7 = 2.5$ ie.

$n^{\log_b a} = n^{\log_b a} = n^{2.5}$ ($n^{\log_b a} > f(n)$..case 3)

$T(n) = \Theta(n^{2.5})$

- Solve $T(n) = 2T(n/2) + n\lg n$ using master's theorem

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

$\log_b a = \log_2 2 = 1$ ie.

$n^{\log_b a} = n^{\log_b a} = n^1$ ($n^{\log_b a} < f(n)$ but $f(n)$ is not polynomially larger than $n^{\log_b a}$, so this problem cannot be solved by master's theorem