Master Theorem

Master theorem

- Master theorem is used to determine running time of algorithms in terms of asymptotic notations.
- The master theorem is a formula for solving recurrences of the form $T(n) = aT(n/b) + f(n)$, where $a \geq 1$ and $b > 1$ and $f(n)$ is asymptotically positive. (Asymptotically positive means that the function is positive for all sufficiently large n.)
- This recurrence describes an algorithm that divides a problem of size n into a subproblems, each of size n/b, and solves them recursively.

Master Theorem - Proof

The theorem is as follows:

Master Theorem:

Let $a \geq 1$ and $b > 1$ be constants, let $f(n)$ be a function, and let $T(n)$ be defined on the nonnegative integers by the recurrence

$$T(n) = aT(n/b) + f(n) ,$$

where we interpret n/b to mean either $\lfloor n/b \rfloor$ or $\lceil n/b \rceil$. Then $T(n)$ has the following asymptotic bounds:

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

Master Theorem

- The master theorem compares the function $n^{\log_b a}$ to the function f(n). Intuitively, if $n^{\log_b a}$ is larger (by a polynomial factor), then the solution is $T(n) = \Theta(n^{\log_b a})$. If f(n) is larger (by a polynomial factor), then the solution is $T(n) = \Theta(f(n))$. If they are the same size, then we multiply by a logarithmic factor.
- Be warned that these cases are not exhaustive – for example, it is possible for f(n) to be asymptotically larger than $n^{\log_b a}$, but not larger by a polynomial factor (no matter how small the exponent in the polynomial is). For example, this is true when $f(n) = n^{\log_b a} \log n$. In this situation, the master theorem would not apply, and you would have to use another method to solve the recurrence.

Master Theorem

Master's theorem solves recurrence relations of the form-

$$T(n) = a\,T\left(\frac{n}{b}\right) + \theta\left(n^k \log^p n\right)$$

Master's Theorem

Case-01:

If $a > b^k$, then $T(n) = \theta\,(n^{\log_b a})$

Case-02:

If $a = b^k$ and

If $p < -1$, then $T(n) = \theta\,(n^{\log_b a})$

If $p = -1$, then $T(n) = \theta\,(n^{\log_b a}.\log^2 n)$

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

Case-03:

If $a < b^k$ and

If $p < 0$, then $T(n) = O\,(n^k)$

If $p >= 0$, then $T(n) = \theta\,(n^k \log^p n)$

Master Theorem

<u>Example 1:</u>

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

We compare the given recurrence relation with

$T(n) = aT(n/b) + \theta\ (n^k \log^p n)$.

Then, we have-

$a = 3, b = 2, k = 2, p = 0$

Now, $a = 3$ and $b^k = 2^2 = 4$.

Clearly, $a < b^k$.

So, we follow case- 03.

Since $p = 0$, so we have-

$T(n) = \theta\ (n^k \log^p n)$

$T(n) = \theta\ (n^2 \log^0 n)$

Thus, $\mathbf{T(n) = \theta\ (n^2)}$

<u>Example 2:</u>

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

We compare the given recurrence relation with

$T(n) = aT(n/b) + \theta\ (n^k \log^p n)$.

Then, we have-

$a = 2, b = 2, k = 1, p = 1$

Now, $a = 2$ and $b^k = 2^1 = 2$.

Clearly, $a = b^k$.

So, we follow case-02.

Since $p = 1$, so we have-

$T(n) = \theta\ (n^{\log_b a} . \log^{p+1} n)$

$T(n) = \theta\ (n^{\log_2 2} . \log^{1+1} n)$

$\mathbf{T(n) = \theta\ (n\log^2 n)}$

Master Theorem

<u>Example 3</u>

$T(n) = 8T(n/4) – n^2 \log n$

- The given recurrence relation does not correspond to the general form of Master's theorem.
- So, it can not be solved using Master's theorem.

<u>Example 4</u>

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

- We write the given recurrence relation as $T(n) = 3T(n/3) + n$.
- This is because in the general form, we have θ for function f(n) which hides constants in it.
- Now, we can easily apply Master's theorem.

We compare the given recurrence relation with $T(n) = aT(n/b) + \theta (n^k \log^p n)$.
Then, we have-

$a = 3, b = 3, k = 1, p = 0$

Now, $a = 3$ and $b^k = 3^1 = 3$.
Clearly, $a = b^k$.
So, we follow case-02.

Since $p = 0$, so we have-

$T(n) = \theta (n^{\log_b a} . \log^{p+1} n)$

$T(n) = \theta (n^{\log_3 3} . \log^{0+1} n)$

$T(n) = \theta (n^1 . \log^1 n)$

$\mathbf{T(n) = \theta (n \log n)}$

Master Theorem

Exercise Problems:

1. T(n)=4T(n/2)+n2 =⇒T(n)=Θ(n2logn)(Case2)
2. T(n)=T(n/2)+2n =⇒Θ(2n)(Case3)
3. T (n) = 2nT (n/2) + nn =⇒ Does not apply (a is not constant)
4. T(n)=16T(n/4)+n=⇒T(n)=Θ(n2)(Case1)
5. T(n)=2T(n/2)+nlogn=⇒T(n)=nlog2n(Case2)