Master Theorem

strong text# Master Theorem

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

Master Theorem applies when your problem is in this format, and has 3 cases you can use to solve for Time Complexity.

Case 1: $f(n)$ is smaller than $n^{log_b(a)}$, in which case
$f(n) = O(n^{log_b(a)})$
If this is true, then your answer is:
$T(n) = \Theta(n^{log_b a})$

Case 2: $f(n)$ is equal to $n^{log_b(a)}$, in which case
$f(n) = \Theta(n^{log_b(a)})$
If this is true, then your answer is:
$T(n) = \Theta f(n) logn$

Case 3: $f(n)$ is greater than $n^{log_b(a)}$, in which case
$f(n) = \Omega(n^{log_b(a)})$
If this is true, then your answer is:
$T(n) = \Theta(f(n))$

Note for Cases 2 and 3, you have to supply f(n) in your answer

The master theorem does not identify the upper and lower bounds, it only identifies the asymptotic tight bound

Example:
$T(n) = 2T\frac{n}{2} + n$

In this example, a=2, b=2, $log_b a$ = $log_2 2$ = 1
So, $n^{lob_b a} = n^{log_2 2}$ = n
$f(n) = n$
So, $n^{log_b a} = n = f(n)$
This means that comparing $n^{log_b a}$ with f(n) => f(n) = $\Theta(n^{log_b a})$

Because of that, case 2 can be applied, so our answer is
T(n) = $\Theta(nlogn)$

Example 2

$$T(n) = 2T\left(\frac{n}{2}\right) + n^2$$

Here, $a = 2$, $b = 2$, $\log_2 2 = 1$
$=> n^{\lg_b a} = n^1 = n$
$Also, f(n) = n^2$
$=> f(n) = \Omega(n^{1+\epsilon})$ ($\epsilon = 1$) (comparing $n^{\log_b a}$ with $f(n)$)
Case 3 can be applied if rest of the conditions of case 3 gets satisfied for $f(n)$.

The condition is $af(n/b) \leq cf(n)$ for some $c < 1$ and all sufficiently large n.
For a sufficiently large n, we have,
$af\left(\frac{n}{b}\right) = 2f\left(\frac{n}{2}\right) = 2\frac{n^2}{4} = \frac{n^2}{2} \leq \frac{1}{2}(n^2)$ (for $c = \frac{1}{2}$)

So, the condition is satisfied for $c = \frac{1}{2}$. Thus, $T(n) = \Theta(f(n)) = \Theta(n^2)$

Example 3

$$T(n) = 2T\left(\frac{n}{2}\right) + \sqrt{n}$$

Here, $a = 2$ $b = 2$ $\log_2 2 = 1$
$n^{\log_2 2} = n$
$f(n) = \sqrt{n}$
$f(n) = O(n^{1-\epsilon})$ (Case 2)
$T(n) = \Theta(n)$