Master Theorem

Introduction:

- Master's Thm states:

  Let $a \geq 1$ and $b > 1$ be constants.
  Let $f(n)$ be an asymptotically positive function.
  Let $T(n) \leq a\,T(\frac{n}{b}) + O(f(n))$ be a recurrence relation.

  Then:

  1. If $f(n) = O(n^{\log_b a - \varepsilon})$ for some constant $\varepsilon > 0$, then $T(n) = O(n^{\log_b a})$

  2. If $f(n) = \Theta(n^{\log_b a} \cdot \log^k n)$ for some constant $k \geq 0$, then $T(n) = O(n^{\log_b a} \log^{k+1} n)$

  3. If $f(n) = \Omega(n^{\log_b a + \varepsilon})$ for some constant $\varepsilon > 0$ and f satisfies the regularity condition then $T(n) = O(f(n))$

  The regularity condition states that:
  For some constant $c < 1$ and all sufficiently large n, $a \cdot f(\frac{n}{b}) \leq c \cdot f(n)$

Examples:

1. $T(n) \leq 9\,T(\frac{n}{3}) + n$

   $a = 9$, $b = 3$, $f(n) = n$, $\log_b a = \log_3 9 = 2$
   We see that $n^{\log_b a} = n^2$.
   Hence, $f(n) = O(n^{\log_b a - \varepsilon})$, where $\varepsilon = 1$ in this case.
   $\therefore T(n) = O(n^2)$

2. $T(n) \leq T(\frac{2n}{3}) + 1$

$a=1,\ b=\frac{2}{3},\ f(n)=1,\ \log_b{}^a = 0$

$n^{\log_b a} = n^0$

$= 1$

We see that $f(n) = n^{\log_b a}$. We can see that case 2 applies if $k=0$.

$\therefore\ T(n) = O(n\log n)$

3. $T(n) \leq 2T(\frac{n}{2}) + n\lg(n)$

$a=2,$ $,\ f(n) = n\lg(n),\ \log_b{}^a =$ $\log_2{}^2 = 1$

$b=2$

None of the cases apply here.
While $f(n)$ is larger than $n^{\log_b a}$ or n, it is not polynomially larger.

$\frac{n\lg n}{n} = \lg(n)$

For case 3, we need $f(n) = \Omega(n^{\log_b a + \varepsilon})$

$= \Omega(n^{\log_b a} \cdot n^{\varepsilon})$

and $\lg(n)$ is asymptotically less than n^{ε} for any positive ε.

4. $T(n) \leq 2T(\frac{n}{2}) + O(n)$

Note: This is the run time for the divide and conquer soln to max subarray.

$a=2,\ b=2,\ f(n) = O(n),\ \log_b{}^a = 1$

This would fall under case 2, for $k=0$.

$\therefore\ T(n) = O(n\lg n)$

5. $T(n) \leq 7(T(\frac{n}{2})) + O(n^2)$

Note: This is the run time for Strassen's algo.

$a = 7$, $b = 2$, $f(n) = O(n^2)$, $\log_b a = \log_2 7$
≈ 2.8

The first case applies as $f(n) = O(n^{\log_2 7 - \varepsilon})$ where $\varepsilon = 0.8$.

$\therefore$ $T(n) = O(n^{\log_b a})$
$= O(n^{\lg_2 7})$