Master Method

Way to solve certain recurrences

\[ T(n) = a \cdot T\left(\frac{n}{b}\right) + O(n^d) \]

\[ T(n) \leq C \text{ for } n < n^* \]

\[ a, b, d \text{ don't depend on } n \]

Q: If \( T(n) \) is runtime of an algorithm,

What are \( a, b, d \) in words?

A: \( a \): \# of recursive calls

\( b \): factor by which problem shrinks in recursive call

\( d \): characterizes extra work outside recursive call

Let's Add Up All Work

\[ n \leftarrow \text{Problem size} \]

Input size

\[ n \]

\[ \frac{n}{5} \]

\[ \frac{n}{25} \]

\[ \vdots \]
Q. What is $F$ in terms of $a$, $b$, and $d$?

A) $O(\log n)$
B) $O(\log^2 n)$
C) $O(n^{\log d})$
D) $O(b^{\log n})$

Because at each level, problem size is divided by $b$. $\log n$ is a constant. Number of times $n$ can be divided by $b$ before reaching $1$ takes $\log_b n$. It's also $\log_b n - \log_b c$.

$F = \log_b n - \log_b c$
$F = \log_b n + \text{constant}$
Q: What is the work done at level $K$ (outside of recursive calls & in terms of $a, b, d$)?

- $a^K$ subproblems at level $K$.
- Level $K$ subproblem size: $\frac{n}{b^K}$.
- Work outside of recursive call required to solve 1 subproblem

$\Rightarrow$ Total work $a^K \left( \frac{n}{b^K} \right)^d = \left( \frac{a}{b^d} \right)^K n^d$

Now we add up work done at all levels:

$$T(n) = n^d \left[ \sum_{K=0}^{\log_b n} \left( \frac{a}{b^d} \right)^K \right]$$

Multiplicative Distributive Property
Geometric Series:

\[ \sum_{k=0}^{F} r^k = \begin{cases} 
F+1 & \text{if } r = 1 \\
\frac{1-r^{F+1}}{1-r} & \text{otherwise}
\end{cases} \]
2 cases:

- \( \frac{a}{b^d} = 1 \) \( \Rightarrow \) \( n^d \sum_{k=0}^{\log_{b} n} (\frac{a}{b^d})^k = O(n^d \log_{b} n) \)

- \( \frac{a}{b^d} \neq 1 \) \( \Rightarrow \) \( n^d \sum_{k=0}^{\log_{b} n} (\frac{a}{b^d})^k = n^d \frac{1 - (\frac{a}{b^d})^{\log_{b} n}}{1 - \frac{a}{b^d}} \uparrow \text{constant} = c \)

Look at:

2 cases

- \( (\frac{a}{b^d}) < 1 \) \( \Rightarrow \) \( T(n) = O(n^d) \)

- \( (\frac{a}{b^d}) > 1 \) \( \Rightarrow \) \( \frac{1 - (\frac{a}{b^d})^{\log_{b} n}}{1 - \frac{a}{b^d}} = O\left(\left(\frac{a}{b^d}\right)^{\log_{b} n}\right) \)
\[
\left(\frac{a}{b}\right)^{\log_b n} = \frac{\log n}{\log n} = \frac{a \log n}{\log n} = \frac{a}{b^{\log n}}
\]

\[
\log_{x^2} x = \log_{x} x = 2
\]

\[
T(n) = O\left( n^{\frac{d}{d}} \left( \frac{\log n}{n^{\frac{d}{d}}} \right) \right) = O\left( n^{\log_b a} \right)
\]

\[
T(n) = aT\left( \frac{n}{b} \right) + O(n^d)
\]

\[
T(n) = \begin{cases} 
O(n^{d \log n}) & \text{if } a = b^d \\
O(n^d) & \text{if } a < b^d \\
O(n^{\log_b a}) & \text{if } a > b^d 
\end{cases}
\]

Q: Interpret

- Balance between current work + recursive work.
- Run-time dominated by work outside recursive calls.
- Run-time dominated by work in bottom level of tree.