let $a \geq 1$ and $b > 1$ be constants. Let $f(n)$ be a function and $T(n)$ be defined on nonnegative ints by the recurrence: $T(n) = aT\left(\frac{n}{b}\right) + f(n)$

a: # of child nodes

$\frac{n}{b}$: factor to # of operation at child node

$f(n)$: # operations for merging

$\log_b(n)$: if $b^x = n \Rightarrow \log_b(n) = x$

if algorithm divides a problem of size n into subproblems of size $\frac{n}{b}$ at each step, the # of levels or depth of division until reaching the base case (subproblem of size 1) is $\log_b(n)$

$$\rightarrow n \cdot \frac{1}{b} \cdot \frac{1}{b} \cdots \frac{1}{b} = n \cdot \left(\frac{1}{b}\right)^x = 1$$

$$n = b^x \Rightarrow \log_b(n) = x$$

$\log_b(a)$: c_{crit}, a: # of recursive call at each level

shows growth of subproblems vs the depth of the recursion.

$n^{\log_b a}$:

shows how the work per level of recursion tree scales with n.

It's a critical point for comparing the additional work done at each level of recursion ($f(n)$) to decide on the overall time complexity.

represents amount of work done at each level if the work was evenly distributed across the levels of the recursion tree.

Case 1: $f(n) = O(n^{\log_b(a) - \varepsilon})$, $\varepsilon > 0$

$$T(n) = \Theta(n^{\log_b a})$$

- when the work outside of recursion ($f(n)$) grows polynomially slower than work $n^{\log_b(a)}$, it means that majority of work is done at the last level of recursion tree.
- If $f(n)$ is significantly less than work by recursive calls at deeper levels, the overall complexity is dominated by the deepest level of recursion.

Thus $T(n) = \Theta(n^{\log_b a})$.

Case 2: $f(n) = \Theta(n^{\log_b a} \cdot \log^k n)$, $k \geq 0 \in \mathbb{Z}$

$$T(n) = \Theta(n^{\log_b a} \cdot (\log^{k+1}(n))$$

the work $f(n)$ matches work done by recursive calls, scaled by log factor.

- work to split / recombine problem is comparable to subproblems.

regularity condition (ensures geometric decrease in $f(n)$)

Case 3: $f(n) = \Omega(n^{\log_b(a) + \varepsilon})$ and $af\left(\frac{n}{b}\right) \leq kf(n)$ for large n, $k < 1$

$$T(n) = \Theta(f(n))$$

- work to split / recombine problem dominates subproblems.
- $f(n)$ grows polynomially faster than $n^{\log_b a}$

Master Theorem Worksheet

This is a worksheet to help you master solving recurrence relations using the Master Theorem. For each recurrence, either give the asympotic solution using the Master Theorem (state which case), or else state that the Master Theorem doesn't apply. You should be able to go through these **25** recurrences in **10** minutes.

Problem 1-1. $T(n) = 3T(n/2) + n^2$

$c_{crit} = \log_2 3 < 2 \quad n^{c_{crit}} < n^2 \quad T(n) = \theta(n^2)$

Problem 1-2. $T(n) = 7T(n/2) + n^2$

$n^{\log_2 7} > n^2 \Rightarrow T(n) = \theta(n^{\log_2 7})$

Problem 1-3. $T(n) = 4T(n/2) + n^2$

$n^{\log_2 4} = n^2 \Rightarrow T(n) = \theta(n^2 \cdot \log n)$

Problem 1-4. $T(n) = 3T(n/4) + n \lg n$

$n^{\log_4 3} < n \Rightarrow T(n) = \theta(n \log n)$

Problem 1-5. $T(n) = 4T(n/2) + \lg n$

$n^{\log_2 4} = n^2 > \log n \Rightarrow T(n) = \theta(n^2)$

Problem 1-6. $T(n) = T(n-1) + n$

n depth

linearly and decrementally work

$T(n) = n + n-1 + n-2 \dots + 1$

$= \frac{n}{2}(n) = \frac{n^2}{2}$

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

Problem 1-7. $T(n) = 4T(n/2) + n^2 \lg n$

$n^{\log_2 4} \cdot \log n = n^2 \log n \Rightarrow T(n) = \Theta(n^2 \cdot \log^2 n)$

Problem 1-8. $T(n) = 5T(n/2) + n^2 \lg n$

$n^{\log_2 5} > n^2 \Rightarrow T(n) = \Theta(n^{\log_2 5})$

Problem 1-9. $T(n) = 3T(n/3) + n/\lg n$

$n^1 > \frac{n}{\log n}$, but not polynomially smaller.

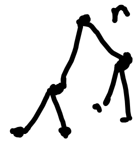
Problem 1-10. $T(n) = 2T(n/4) + c$

$n^{\log_4 2} = n^{1/2} > c \Rightarrow T(n) = \Theta(\sqrt{n})$

Problem 1-11. $T(n) = T(n/4) + \lg n$

$n^{\log_4 1} = n^0 = 1 < \log n \Rightarrow T(n) = \Theta(\log n)$

Problem 1-12. $T(n) = T(n/2) + T(n/4) + n^2$



n^2

$\left(\frac{n}{2}\right)^2 + \left(\frac{n}{4}\right)^2 = \frac{n^2}{4} + \frac{n^2}{16} = \frac{1}{16}(5n^2)$

$2\left(\frac{n}{4}\right)^2 + 2\left(\frac{n}{8}\right)^2 = \frac{n^2}{8} + \frac{n^2}{32} = \frac{5}{32}n^2$

$n^2 + \frac{5}{16}n^2 + \frac{5}{32}n^2 + \dots$

Problem 1-13. $T(n) = 2T(n/4) + \lg n$

$n^{\log_4 2} = n^{0.5} > \log n$

$\Rightarrow T(n) = \Theta(\sqrt{n})$

Problem 1-14. $T(n) = 3T(n/3) + n \lg n$

$n \cdot \log n = n \log n \Rightarrow T(n) = \Theta(n \log^2 n)$

Problem 1-15. $T(n) = 8T((n - \sqrt{n})/4) + n^2$

N/A

Problem 1-16. $T(n) = 2T(n/4) + \sqrt{n}$

$n^{\log_4 2} = n^{0.5} \Rightarrow T(n) = \Theta(\sqrt{n} \log n)$

Problem 1-17. $T(n) = 2T(n/4) + n^{0.51}$

$n^{\log_4 2} < n^{0.51}$ $2 \cdot \left(\frac{n}{4}\right)^{0.51} \le k\, n^{0.51}$ $\Rightarrow T(n) = \Theta(n^{0.51})$

$n^{0.51} \cdot \left(\frac{1}{4}\right)^{0.51} \le k\, n^{0.51}$

Problem 1-18. $T(n) = 16T(n/4) + n!$

$n^{\log_4 16} = n^2 \Rightarrow T(n) = \Theta(n!)$

Problem 1-19. $T(n) = 3T(n/2) + n$

$n^{\log_2 3} > n \Rightarrow T(n) = \Theta(n^{\log_2 3})$

Problem 1-20. $T(n) = 4T(n/2) + cn$

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

Problem 1-21. $T(n) = 3T(n/3) + n/2$

$T(n) = \Theta(n \log n)$

Problem 1-22. $T(n) = 4T(n/2) + n/\lg n$

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

Problem 1-23. $T(n) = 7T(n/3) + n^2$

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

Problem 1-24. $T(n) = 8T(n/3) + 2^n$

$8(2^{n/3}) = 2^{\frac{n}{3} + \frac{9}{3}} = 2^{\frac{n+9}{3}} \leq k\,2^n \Rightarrow T(n) = \Theta(2^n)$

Problem 1-25. $T(n) = 16T(n/4) + n$

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