

Design and Analysis of Algorithms

Tutorial-4

Master Theorem

The Master Theorem applies to recurrences of the following form: $T(n) = aT(n/b) + f(n)$

where $a \geq 1$ and $b > 1$ are constants and $f(n)$ is an asymptotically positive function. There are 3 cases:

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 $f(n) = \Theta(n^{\log_b a} \log^k n)$ with $k \geq 0$, then $T(n) = \Theta(n^{\log_b a} \log^{k+1} n)$.
3. If $f(n) = \Omega(n^{\log_b a + \epsilon})$ with $\epsilon > 0$, and $f(n)$ satisfies the regularity condition, then $T(n) = \Theta(f(n))$. Regularity condition: $af(n/b) \leq cf(n)$ for some constant $c < 1$ and all sufficiently large n .

Practice Problems

For each of the following recurrences, give an expression for the runtime $T(n)$ if the recurrence can be solved with the Master Theorem. Otherwise, indicate that the Master Theorem does not apply.

1. $T(n) = 3T(n/2) + n^2$	12. $T(n) = \sqrt{n}T(n/2) + \log n$
2. $T(n) = 4T(n/2) + n^2$	13. $T(n) = 3T(n/2) + n$
3. $T(n) = T(n/2) + 2^n$	14. $T(n) = 3T(n/3) + \sqrt{n}$
4. $T(n) = 2^n T(n/2) + n^n$	15. $T(n) = 4T(n/2) + cn$
5. $T(n) = 16T(n/4) + n$	16. $T(n) = 3T(n/4) + n \log n$
6. $T(n) = 2T(n/2) + n \log n$	17. $T(n) = 3T(n/3) + n/2$
7. $T(n) = 2T(n/2) + n/\log n$	18. $T(n) = 6T(n/3) + n^2 \log n$
8. $T(n) = 2T(n/4) + n^{0.51}$	19. $T(n) = 4T(n/2) + n/\log n$
9. $T(n) = 0.5T(n/2) + 1/n$	20. $T(n) = 64T(n/8) - n^2 \log n$
10. $T(n) = 16T(n/4) + n!$	21. $T(n) = 7T(n/3) + n^2$
11. $T(n) = 4T(n/2) + \log n$	22. $T(n) = T(n/2) + n(2 - \cos n)$