Recurrence Relations

Prem Nair

$T(1) = d$, $T(n) = aT(n/b) + cn^k$ $(n > 1)$
Assume $n = b^p$ or $p = \log_b n$

$T(b^p) = aT(b^{p-1}) + c(b^p)^k$

$aT(b^{p-1}) = a^2T(b^{p-2}) + ac(b^{p-1})^k$

$a^2T(b^{p-2}) = a^3T(b^{p-3}) + a^2c(b^{p-2})^k$

. . .

$a^{p-2}T(b^2) = a^{p-1}T(b) + a^{p-2}c(b^2)^k$

$a^{p-1}T(b) = a^pT(1) + a^{p-1}c(b)^k$

$T(n) = c(b^p)^k + ac(b^{p-1})^k + a^2c(b^{p-2})^k + \dots + a^{p-2}c(b^2)^k + a^{p-1}c(b)^k + a^pT(1)$

$T(n) = c[(b^p)^k + a(b^{p-1})^k + a^2(b^{p-2})^k + \dots + a^{p-2}(b^2)^k + a^{p-1}(b)^k] + a^pd.$

$$T(1) = d,\ T(n) = T(n/2) + c \quad (n > 1)$$

$$a = 1,\ b = 2,\ k = 0.\ a = b^0$$

$T(n) = c[\ (b^p)^k + a(b^{p-1})^k + a^2(b^{p-2})^k + \ldots + a^{p-2}(b^2)^k + a^{p-1}(b)^k\ ] + a^p d.$

$T(n) = c[\ 1 + 1 + 1 + \ldots + 1 + 1] + d = cp + d = O(p) = O(\log n)$

Master Theorem

$n^k \log_b n = \log_2 n$

Hence by Master Theorem, we have $\Theta(\log_2 n)$

$$T(1) = d, \; T(n) = 2T(n/2) + c \qquad (n > 1)$$
$$a = 2, \; b = 2, \; k = 0. \; a > b^0$$

$T(n) = c[\, (b^p)^k + a(b^{p-1})^k + a^2(b^{p-2})^k + \ldots + a^{p-2}(b^2)^k + a^{p-1}(b)^k \,] + a^p d.$

$T(n) = c[\, 1 + 2 + 2^2 + \ldots + 2^{p-2} + 2^{p-1}] + 2^p d = O(2^p) = O(n)$

Master Theorem

$\log_b a = \log_2 2 = 1.$

Hence by Master Theorem, we have $\Theta(n)$

$$T(1) = d, T(n) = 4T(n/2) + c \quad (n > 1)$$
$$a = 4, b = 2, k = 0. \ a > b^0$$

$T(n) = c[\, (b^p)^k + a(b^{p-1})^k + a^2(b^{p-2})^k + . . . + a^{p-2}(b^2)^k + a^{p-1}(b)^k\,] + a^p d.$

$T(n) = c[\, 1 + 4 + 4^2 + . . . + 4^{p-2} + 4^{p-1}] + 4^p d = O(4^p) = O((2^p)^2) = O(n^2).$

Master Theorem

$\log_b a = \log_2 4 = 2.$

Hence by Master Theorem, we have $\Theta(n^2)$.

$T(1) = d,\ T(n) = T(n/2) + cn \quad (n > 1)$
$a = 1,\ b = 2,\ k = 1.\ a < b^1$

$T(n) = c[\ (b^p)^k + a(b^{p-1})^k + a^2(b^{p-2})^k + \ldots + a^{p-2}(b^2)^k + a^{p-1}(b)^k\ ] + a^p d.$

$T(n) = c[\ (2^p) + (2^{p-1}) + (2^{p-2}) + \ldots + (2^2) + (2)\ ] + d < 2n + d = O(n)$

Master Theorem

$k = 1$

Hence by Master Theorem, we have $\Theta(n^k)$. That is, $\Theta(n)$.

$T(1) = d, T(n) = 2T(n/2) + cn \quad (n > 1)$
$a = 2, b = 2, k = 1. a = b^1$

$T(n) = c[(b^p)^k + a(b^{p-1})^k + a^2(b^{p-2})^k + . . . + a^{p-2}(b^2)^k + a^{p-1}(b)^k] + a^p d.$

$T(n) = c[(2^p) + 2(2^{p-1}) + 2^2(2^{p-2}) + . . . + 2^{p-2} (2^2) + 2^{p-1}(2)] + d < O(n\log n)$

Master Theorem

$k = 1$

Hence by Master Theorem, we have $\Theta(n^k)$. That is, $\Theta(n\log n)$.

$$T(1) = d, T(n) = 4T(n/2) + cn \quad (n > 1)$$

$$a = 4, b = 2, k = 1. \; a = b^2$$

$T(n) = c[\,(b^p)^k + a(b^{p-1})^k + a^2(b^{p-2})^k + \ldots + a^{p-2}(b^2)^k + a^{p-1}(b)^k\,] + a^p d.$

$T(n) = c[\,(2^p) + 4(2^{p-1}) + 4^2(2^{p-2}) + \ldots + 4^{p-2}(2^2) + 4^{p-1}(2)\,] + d = O(4^p)$

$= O((2^p)^2) = O(n^2).$

Master Theorem

$k = 1$

$\log_b a = \log_2 4 = 2.$

Hence by Master Theorem, we have $\Theta(n^2)$.

The Master Theorem

For recurrences that arise from Divide-And-Conquer algorithms (like Binary Search), there is a general formula that can be used.

Theorem. Suppose $T(n)$ satisfies

$$T(n) = \begin{cases} d & \text{if } n = 1 \\ aT(\lceil \frac{n}{b} \rceil) + cn^k & \text{otherwise} \end{cases}$$

where k is a nonnegative integer and a, b, c, d are constants with $a > 0, b > 1, c > 0, d \geq 0$. Then

$$T(n) = \begin{cases} \Theta(n^k) & \text{if } a < b^k \\ \Theta(n^k \log n) & \text{if } a = b^k \\ \Theta(n^{\log_b a}) & \text{if } a > b^k \end{cases}$$