Design and Analysis of Algorithms I

Master Method

Examples

The Master Method

If $T(n) \leq aT\left(\frac{n}{b}\right) + O(n^d)$

then

$$T(n) = \begin{cases} O\left(n^d \log n\right) & \text{if } a = b^d \quad \text{(Case 1)} \\ O(n^d) & \text{if } a < b^d \quad \text{(Case 2)} \\ O(n^{\log_b a}) & \text{if } a > b^d \quad \text{(Case 3)} \end{cases}$$

Example #1

Merge Sort

$a = 2$

$b = 2$

$d = 1$

$b^d = 2 \Rightarrow$ Case 1

$\Rightarrow T(n) \leq O(n^d \log n) = O(n \log n)$

Where are the respective values of a, b, d for a binary search of a sorted array, and which case of the Master Method does this correspond to?

- 1, 2, 0 [Case 1]
- 1, 2, 1 [Case 2]
- 2, 2, 0 [Case 3]
- 2, 2, 1 [Case 1]

$a = b^d \Rightarrow T(n) = O(n^d \log n) = O(\log n)$

Example #3

Integer Multiplication Algorithm #1

$a = 4$

$b = 2$

$d = 1$ — $b^d = 2 < a$ (Case 3)

$$\Rightarrow T(n) = O(n^{\log_b a}) = O(n^{\log_2 4})$$

$$= O(n^2)$$

same as grade-school algorithm :(

Tim Roughgarden

Where are the respective values of a, b, d for Gauss's recursive integer multiplication algorithm, and which case of the Master Method does this correspond to?

- $2, 2, 1$ [Case 1]
- $3, 2, 1$ [Case 1]
- $3, 2, 1$ [Case 2]
- $3, 2, 1$ [Case 3]

$a = 3, b^d = 2 \quad a > b^d$ (Case 3)

$\Rightarrow T(n) = O(n^{\log_2 3}) = O(n^{1.59})$

better than the grade-school algorithm!

Example #5

Strassen's Matrix Multiplication Algorithm

$a = 7$

$b = 2$

$d = 2$

$b^d = 4 < a$

(Case 3)

$\Rightarrow T(n) = O(n^{\log_2 7}) = O(n^{2.81})$

$\Rightarrow$ beats the naive iterative algorithm!

Example #6

Fictitious recurrence

$T(n) \leq 2T\left(\frac{n}{2}\right) + O(n^2)$

$\Rightarrow a = 2$

$\left.\begin{array}{l} b = 2 \\ d = 2 \end{array}\right\} b^d = 4 > a$ (Case 2)

$\Rightarrow T(n) = O(n^2)$