Theorem (Master Theorem)

Let $T : \mathbb{N} \longrightarrow \mathbb{R}^+$ *be a function. Suppose that*

$$T(n) = \begin{cases} 1 & \text{if } n = 1, \\ a \cdot T(n/b) + n^d & \text{if } n > 1, \end{cases}$$

where $a \geq 1$*,* $b > 1$ *and* $d \geq 0$*.*

- *If* $d > \log_b(a)$*, then* $T(n) = O\left(n^d\right)$*.*
- *If* $d = \log_b(a)$*, then* $T(n) = O\left(n^d \log(n)\right)$*.*
- *If* $d < \log_b(a)$*, then* $T(n) = O\left(n^{\log_b(a)}\right)$*.*

Merge sort: $T(n) = 2T(n/2) + n^1$

Merge sort: $T(n) = 2T(n/2) + n^1$

$a = 2,\ b = 2,\ d = 1$

Merge sort: $T(n) = 2T(n/2) + n^1$

$a = 2,\ b = 2,\ d = 1$ $d = \log_b(a)$

Merge sort: $T(n) = 2T(n/2) + n^1$

$a = 2,\ b = 2,\ d = 1$ $\qquad d = \log_b(a)$

$$T(n) = O(n^1 \log(n)) = O(n \log(n))$$

Merge sort: $T(n) = 2T(n/2) + n^1$

$a = 2,\ b = 2,\ d = 1$ $d = \log_b(a)$

$$T(n) = O(n^1 \log(n)) = O(n \log(n))$$

Strassen: $T(n) = 7T(n/2) + n^2$

Merge sort: $T(n) = 2T(n/2) + n^1$

$a = 2,\ b = 2,\ d = 1$ $d = \log_b(a)$

$$T(n) = O(n^1 \log(n)) = O(n \log(n))$$

Strassen: $T(n) = 7T(n/2) + n^2$

$a = 7,\ b = 2,\ d = 2$

Merge sort: $T(n) = 2T(n/2) + n^1$

$a = 2,\ b = 2,\ d = 1$ $d = \log_b(a)$

$$T(n) = O(n^1 \log(n)) = O(n \log(n))$$

Strassen: $T(n) = 7T(n/2) + n^2$

$a = 7,\ b = 2,\ d = 2$ $d < \log_b(a)$

Merge sort: $T(n) = 2T(n/2) + n^1$

$a = 2,\ b = 2,\ d = 1$ $\qquad d = \log_b(a)$

$$T(n) = O(n^1 \log(n)) = O(n \log(n))$$

Strassen: $T(n) = 7T(n/2) + n^2$

$a = 7,\ b = 2,\ d = 2$ $\qquad d < \log_b(a)$

$$T(n) = O\left(n^{\log_2(7)}\right)$$

Merge sort: $T(n) = 2T(n/2) + n^1$

$a = 2,\ b = 2,\ d = 1$ $\quad d = \log_b(a)$

$$T(n) = O(n^1 \log(n)) = O(n \log(n))$$

Strassen: $T(n) = 7T(n/2) + n^2$

$a = 7,\ b = 2,\ d = 2$ $\quad d < \log_b(a)$

$$T(n) = O\left(n^{\log_2(7)}\right)$$

Binary Search: $T(n) = T(n/2) + n^0$

Merge sort: $T(n) = 2T(n/2) + n^1$

$a = 2,\ b = 2,\ d = 1$ $\quad d = \log_b(a)$

$$T(n) = O(n^1 \log(n)) = O(n \log(n))$$

Strassen: $T(n) = 7T(n/2) + n^2$

$a = 7,\ b = 2,\ d = 2$ $\quad d < \log_b(a)$

$$T(n) = O\left(n^{\log_2(7)}\right)$$

Binary Search: $T(n) = T(n/2) + n^0$

$a = 1,\ b = 2,\ d = 0$

Merge sort: $T(n) = 2T(n/2) + n^1$

$a = 2,\ b = 2,\ d = 1$ $d = \log_b(a)$

$$T(n) = O(n^1 \log(n)) = O(n \log(n))$$

Strassen: $T(n) = 7T(n/2) + n^2$

$a = 7,\ b = 2,\ d = 2$ $d < \log_b(a)$

$$T(n) = O\left(n^{\log_2(7)}\right)$$

Binary Search: $T(n) = T(n/2) + n^0$

$a = 1,\ b = 2,\ d = 0$ $d = \log_b(a)$

Merge sort: $T(n) = 2T(n/2) + n^1$

$a = 2,\ b = 2,\ d = 1$ $\quad d = \log_b(a)$

$$T(n) = O(n^1 \log(n)) = O(n \log(n))$$

Strassen: $T(n) = 7T(n/2) + n^2$

$a = 7,\ b = 2,\ d = 2$ $\quad d < \log_b(a)$

$$T(n) = O\left(n^{\log_2(7)}\right)$$

Binary Search: $T(n) = T(n/2) + n^0$

$a = 1,\ b = 2,\ d = 0$ $\quad d = \log_b(a)$

$$T(n) = O(n^0 \log(n)) = O(\log(n))$$