Master's Theorem

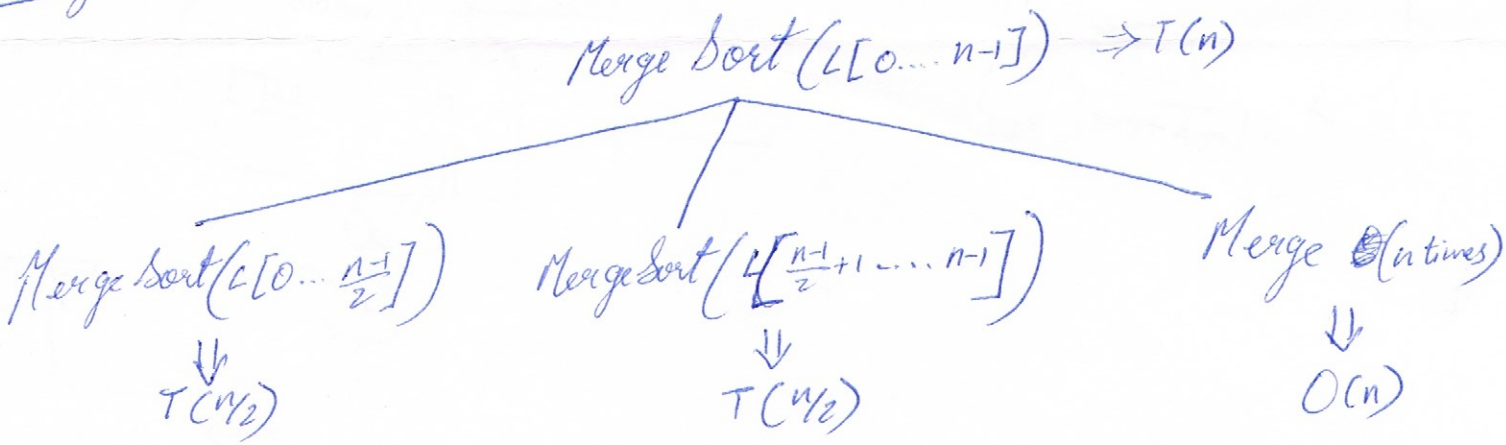
$$T(n) = aT(n/b) + f(n)$$
$$T(1) = c$$

$a \geq 1$, $b \geq 2$, $c > 0$

if $f(n) = O(n^d)$; $d \geq 0$, then.

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

Merge Sort



Thus → $T(n) = 2T(n/2) + O(n)$

$T(1) = O(1)$

Applying Master's theorem; $a=2$, $b=2$, $d=1$

$a = 2$, $b^d = 2^1 = 2$ ⇒ $a = b^d$

Thus $\underline{T(n) = O(n^d \log n) = O(n \log n)}$