

/ Wed, 7th October / Divide & Conquer.

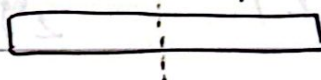
* Short history

Date	Authors	Runtime
Ancient		$\Theta(n^2)$
1962	Karatsuba	$\Theta(n \log_2^3) \sim \Theta(n^{1.58...})$
1966	Toom - Cook	$\Theta(n^{1+\epsilon})$
1971	Schönhage - Strassen	$\Theta(n \log n \log \log n)$
2007 - 2018	Harvey - Heven - Lecerc	$\Theta(n \log n 2^{O(\log^* n)})$
2019	Harvey - Heven	$\Theta(n \log n)$
2026	Open AI	$O(n (\log n)^{1+k}) \quad (k = \frac{1}{2^{182}})$

① Merge - Sort Algorithm

* Sort an array of length n : $A[0, \dots, n-1]$

* Algorithm: Merge-Sort ($A[0, \dots, n-1]$)

If $n \leq 1$: 

return A

else :

$L = A[0 : \frac{n}{2}]$

$R = A[\frac{n}{2} : n]$

$L = \text{MergeSort}(L)$

$R = \text{MergeSort}(R)$

return Merge(L, R)

* Algorithm : Merge(L, R):

→ runtime: $m+n$

out = []

$L_idx, R_idx = 0$

while $\text{len}(\text{out}) \leq \text{len}(L) + \text{len}(R)$:

if $(R_idx == \text{len}(R)) \vee (L[L_idx] \leq R[R_idx])$:

$L_idx < \text{len}(L)$ &&

add $L[L_idx]$ to out

L_idx++

else

add $R[R_idx]$ to out ; R_idx++

return out

- Let $T(n)$ be the runtime of Merge Sort on an array of length n

$$T(0) = T(1) = \Theta(1)$$

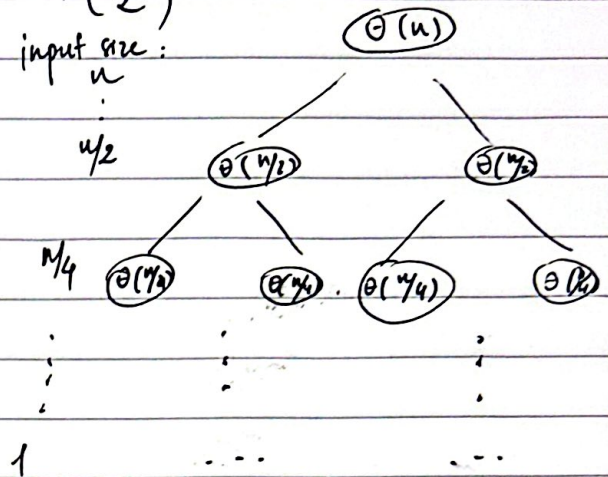
$$T(n) = \Theta(n) + 2T\left(\frac{n}{2}\right)$$

* Height: $\log_2 n$

Total work per level: $\Theta(n)$

$$T(n) = \Theta(n \log n)$$

input size: n



② Quick-Sort Algorithm

* Quick-Sort($A[0, \dots, n-1]$)

if $n \leq 1$:

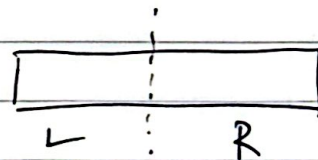
return A

else:

* do sth to $L \rightarrow$ get all "small things" in L s.t. $L \leq R$
 * do sth to $R \rightarrow$ get all "big things" in R

Quick-Sort(L)

Quick-Sort(R)



* Pick an item in A

put everything smaller than it in L
 " bigger " in R

If chosen badly,
 $T(n) = \Theta(n) + T(n-1)$
 $= \Theta(n^2)$