

Θ, Ω, O
$$\underline{\mu_n^r + q_n} = \underline{O(n^r)}$$
$$f_{(n)} = o(g(n)) : \underbrace{\forall_c}_{\substack{\leftarrow \quad \rightarrow \\ \left(\begin{smallmatrix} \frac{1}{c} & \frac{1}{c} \\ \vdots & \vdots \end{smallmatrix} \right)}} \exists n. \quad \forall n_{\geq n}. \quad \underbrace{f_{(n)}}_{\substack{\leftarrow \quad \rightarrow \\ \left(\begin{smallmatrix} \frac{1}{c} & \frac{1}{c} \\ \vdots & \vdots \end{smallmatrix} \right)}} : \underbrace{g(n)}_{\substack{\leftarrow \quad \rightarrow \\ \left(\begin{smallmatrix} \frac{1}{c} & \frac{1}{c} \\ \vdots & \vdots \end{smallmatrix} \right)}} = o(g(n)) : \underbrace{\frac{0}{0}}_{\substack{\downarrow \\ 0}} : o : \omega$$
$$f(n) = \omega(g(n)) : \forall c \exists n. \forall n \geq n. f(n) \geq c g(n)$$

$\omega - \Omega - \Theta - \text{مترددات} : 0$

	ثابت	خطی	درجه دوم	درجه سوم	درجه چهارم
$O(1)$	$O(\log n)$	$O(n)$	$O(n \log n)$	$O(n^2)$	$O(n^3)$
ثابت	مربعی	مکعبی			
	$O(n^k)$	$O(n^n)$	$O(n!)$	$O(n^n)$	
	چند جمله ای	توانی			

$$\begin{aligned} f_{(n)} &= O(g_{l(n)}) \\ f_{r(n)} &= O(g_{r(n)}) \end{aligned} \rightsquigarrow f_{l(n)} \times f_{r(n)} = O(g_{l(n)} \times g_{r(n)})$$
$$\begin{aligned} f_l(n) &= O(g_l(n)) \\ f_r(n) &= O(g_r(n)) \\ f_l(n) + f_r(n) &= O(\max[g_l(n), g_r(n)]) \end{aligned}$$
$$\left. \begin{array}{l} f(n) = O(n) \\ g(n) = O(\log n) \end{array} \right\} \quad f(n) + g(n) = O(n)$$

$O(1)$
 $O(1) = 5$ \rightarrow $\{ \text{array of 5 elements} \}$ \rightarrow $O(1)$
 $5 \times O(1) = O(1)$
 $\{ \text{for}(i=1 \rightarrow n) \}$ \rightarrow $O(n)$
 $\{ \text{for}(i=1 \rightarrow n) \}$ \rightarrow $O(n)$
 $\{ \text{for}(i=1 \rightarrow n) \}$ \rightarrow $O(n)$
 $\text{bin_search}() \rightarrow O(\log n)$
 $\text{seq_search}() \rightarrow O(n)$
 $O(n) \times O(\log n) = O(n \log n)$
 $O(n)$

$O(n \log n)$

کلیں روائے بازلسی : روس جاملندری، امسترا و درخت بازلس

Master thm

مثال ۱) زین اجرای کد زیر چقدر راست؟ (جستجوی دودویی) $n = ۱۲$ $x \leftarrow \text{binsearch}(\text{low}, \text{high})$

$T(n) = O(1) + T(n/2)$

```

O(1) {
    if (low > high)
        return
    mid = (low + high) / 2
}

```

n

$\left(\begin{array}{l} \text{if } (A[\text{mid}] > x) \\ \text{bin} \dots (\text{low}, \text{mid}-1) \end{array} \right)$

$\left(\begin{array}{l} \text{if } (A[\text{mid}] < x) \\ \text{bin} \dots (\text{mid}+1, \text{high}) \end{array} \right)$

$O(1) \leftarrow \text{return (mid)}$

$$\leftarrow T(n) = T(n/4) + O(1) \quad T(1) = O(1)$$
$$T(n) = O(\log n)$$
$$T(n) = T\left(\frac{n}{f}\right) + O(1)$$
$$T(n) = T\left(\frac{n}{2}\right) + O(1) + O(1)$$
$$= T(\frac{n}{\lambda}) + O(1) + O(1) + O(1)$$

...

$$\underbrace{T(n) + O(1) + \dots + O(1)}_{O(\log n)} \rightarrow O(\log n)$$

مسال ۲) $T(1) = O(1) \quad T(n) = T(\frac{n}{2}) + O(n)$

آیا این روش صحیح است؟

$$\begin{aligned} T(n) &= T(\frac{n}{2}) + O(n) \\ &= T(\frac{n}{2}) + O(n) + O(n) \\ &= T(\frac{n}{2}) + O(n) + O(n) + O(n) \\ &\vdots \end{aligned}$$

$$\overleftrightarrow{O(n \log n)} = T(1) + \underbrace{O(n) + \dots + O(n)}_{\log n}$$

$T(1) = O(1) \ll C_1 \quad T(n) = T(\frac{n}{2}) + O(n)$

$1 \leftarrow n$

$$T(n) \ll T(\frac{n}{2}) + C_1 n$$

$$\begin{aligned} \leftarrow T(n) &\ll T(\frac{n}{2}) + C_1 n \\ &\ll T(\frac{n}{2}) + \frac{C_1 n}{2} + C_1 n \\ &\ll T(\frac{n}{2}) + \frac{C_1 n}{2} + \frac{C_1 n}{2} + C_1 n \\ &\vdots \\ &\ll \underbrace{T(1)}_{C_1} + \underbrace{2}_{C_1} + \underbrace{2}_{C_1} + \dots + \underbrace{\frac{n}{2}}_{C_1} + \underbrace{n}_{C_1} \\ &\ll C_1 + C_1 \underbrace{\left[1 + 2 + 2 + \dots + \frac{n}{2} + n \right]}_{\ll 2n} \\ &\ll C_1 + 2n C_1 = O(n) \end{aligned}$$

مسال ۳) مرتب سازی ادغامی Merge-sort : اولین روشی که در $O(n \log n)$ بهتر از n^2 است.

مساله: آرایه A شامل n عدد داده شده است. A را مرتب کنید.

زیرمساله (ادغام): دو آرایه A_1 و A_2 به صورتی مرتب شده داده شده

$\frac{n}{2}$ $\frac{n}{2}$ A_1 و A_2 ادغام

$A_1 = [3, 4, 8, 10]$ $A_2 = [2, 5, 9, 11]$

$\rightarrow [2, 3, 4, 5, 8, 9, 10, 11]$

$O(n)$ زمان اجرا

ادغام

مرتب سازی ادغامی: A : n

۱- آرایه A را به دو قسمت مساوی با اندازه $\frac{n}{2}$ تقسیم کن A_1 و A_2

۲- A_1 و A_2 را به صورتی با زبستگی مرتب کنید.

۳- A_1 و A_2 را ادغام کنید

محدوده آرایه

$m_sort(low, high)$

$O(1)$

if (low \geq high)

return

mid = (low + high) / 2

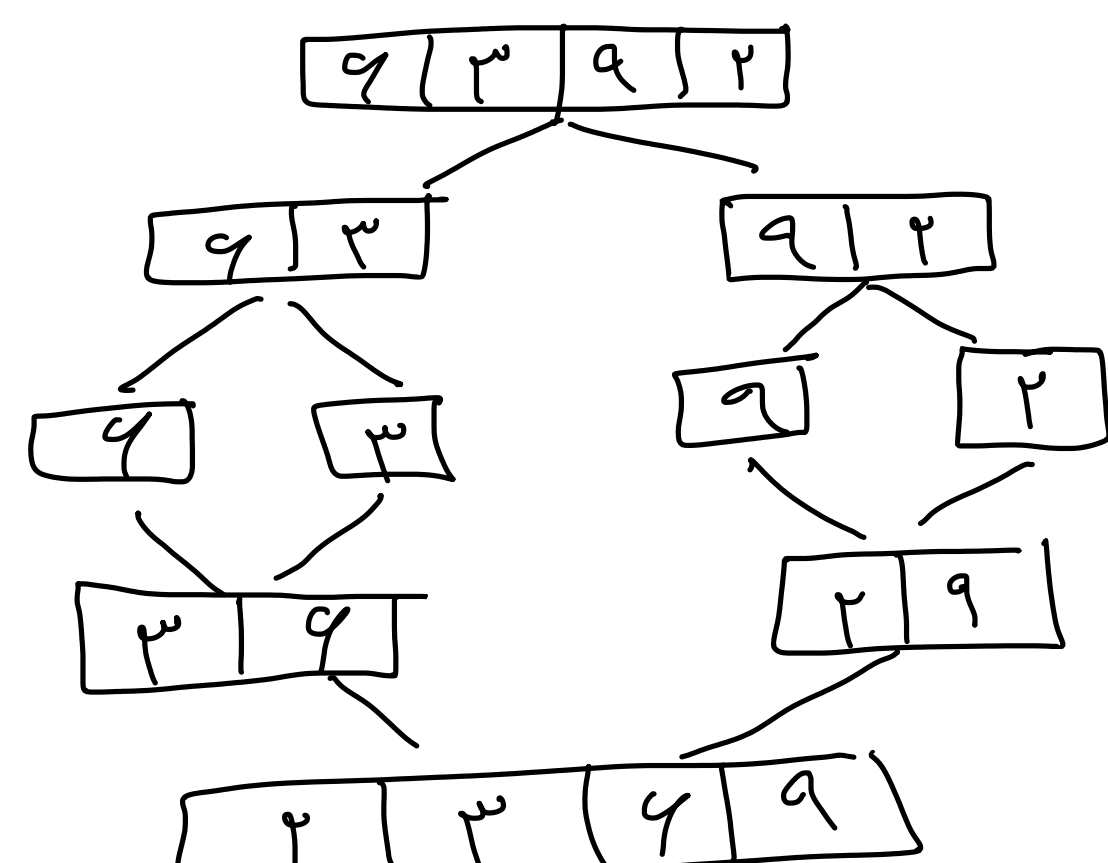
m_sort(low, mid)

m_sort(mid + 1, high)

merge(low, mid, mid + 1, high)

$O(n)$

}



$T(n)$