Divide & conquer

problem - divided into several subproblems

$T(n) = aT(n/b) + f(n)$ - order of growth

Recurrence is called divide & conquer recurrence

$f(n) \in O(n^d)$ $d > 0$

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

① **Mergesort :**

↳ merging operation repeated until one of the 2 given arrays is exhausted.

$c(n) = 2c(n/2) + C_{merge}(n)$ $n > 1,$

$c(1) = 0.$

$C_{worst}(n) = n\log_2 n - n + 1$ } worst case when $n = 2^k$.

complexity : $\boxed{O(n \log_2 n)}$

② **quicksort**

worst : $O(n^2)$

Best, Average : $O(n \log_2 n)$ $n = 2^k$

$C_{best}(n) = 2C_{best}(n/2) + n$ $n > 0$

$C_{best}(1) = 0,$ $C_{avg}(0) = 0$ $C_{avg}(1) = 0$

```
       If (i<j)                 i++
       If (j>i)                 j--
ack.   If (A[i] > A[j])         swap A[i], A[j]
       If (A[i] > p)            swap p, A[j]
```

BC : $T(n) = 2T(n/2) + n$ $O(n\log_2 n)$

WC : $T(n) = T(n-1) + n$ $O(n^2)$

* Strassen's Matrix

$c = a * b = c_2 10^2 + c_1 10^1 + c_0$

mult of large integers

$c_2 = a_1 * b_1$ - product of 1st digits

$c_0 = a_0 * b_0$ - " " 2nd.

$c_1 = (a_1 + a_0) * (b_1 + b_0) - (c_2 + c_0)$

$A(n) = 3A(n/2) + n$ $n > 1$, $A(1) = 1$

$A(n) \in \boxed{O(n^{\log_2 3})}$ || $M(n) = n^{1.585}$

$M(n) = 3M(n/2)$

- STASSENS :

$A(n) = 7A(n/2) + 18(n/2)^2$ $n > 1$

$\Rightarrow \boxed{O(n^{\log_2 7})}$ $A(1) = 0$

$M(n) = n^{2.807}$ | $M(n) = 7M(n/2)$ $n > 1$

$M(1) = 1$

$$\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} * \begin{bmatrix} b_{00} & b_{01} \\ b_{10} & b_{11} \end{bmatrix}$$

$$= \begin{bmatrix} m_1 + m_4 - m_5 + m_7 & m_3 + m_5 \\ m_2 + m_4 & m_1 + m_3 - m_2 + m_6 \end{bmatrix}$$

$m_1 = (a_{00} + a_{11}) * (b_{00} + b_{11})$

$m_2 = (a_{10} + a_{11}) * b_{00}$

$m_3 = a_{00} * (b_{01} - b_{11})$

$m_4 = a_{11} * (b_{10} - b_{00})$

$m_5 = (a_{00} + a_{01}) * b_{11}$

$m_6 = (a_{10} - a_{00}) * (b_{00} + b_{01})$

$m_7 = (a_{01} - a_{11}) * (b_{10} + b_{11})$

* Binary search

$C_{worst}(n) = C_{worst}(\lfloor \frac{n}{2} \rfloor + 1)$ $\forall n > 1$

masters $\Rightarrow$ $\boxed{C_{worst}(n) = O(\log_2 n)}$