# CS 170

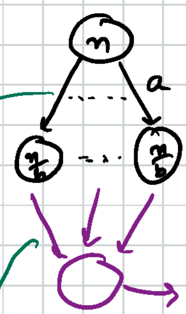
## Lecture 2

- use iClicker

Sanjam Garg.

join.iclicker.com/RGWR

## Divide & Conquer




solution for the original problem.

e.g. $b = 2$

$$T(n) = a\,T\left(\tfrac{n}{b}\right) + O(n^d)$$

$x = $ [ $x_L$ | $x_R$ ] $\qquad y = $ [ $y_L$ | $y_R$ ]

$$x = x_L 2^{\frac{n}{2}} + x_R \qquad y = y_L 2^{\frac{n}{2}} + y_R$$

$$x \times y = \underset{①}{x_L y_L}\, 2^n + (\underset{②}{x_L y_R} + \underset{③}{x_R y_L})\, 2^{\frac{n}{2}} + \underset{④}{x_R y_R}$$

$\rightarrow a = 4$

$b = 2$

$O(n) \rightarrow d = 1$

MULTIPLY $(x, y)$

Input: $x, y \rightarrow$ n-bit positive integers

Output: $x \times y$

$\log_2 4 = 2 \Leftrightarrow 2^2 = 4$

$\log_2 8 = 3 \Leftrightarrow 2^3 = 8$

```
MUL(x, y)
    if n=1    return x×y
    xL, xR = left ⌈n/2⌉, right ⌊n/2⌋ bits of x
    yL, yR = left ⌈n/2⌉, right ⌊n/2⌋ bits of y
    P1, P2, P3, P4 = MUL(xL, yL), MUL(xL, yR), MUL(xR, yL), MUL(xR, yR)
    return  P1 2^n + (P2+P3) 2^(n/2) + P4
```

$a = 4 \quad b = 2$

$d = 1$

$d = 1 < 2 = \log_b a$

$O(n^{\log_b a})$

$= O(n^2)$

$$T(n) = \overset{a}{4}\, T\left(\frac{n}{\underset{b}{2}}\right) + O(n^{\overset{d}{1}}) \longrightarrow O(n^2)$$

$$P_2 + P_3 = x_L y_R + x_R y_L = (x_L + x_R)(y_L + y_R) - \underbrace{x_L y_L}_{P_1} - \underbrace{x_R y_R}_{P_4}$$

```
MUL(x, y)
    if n=1    return x×y
    xL, xR = left ⌈n/2⌉, right ⌊n/2⌋ bits of x
    yL, yR = left ⌈n/2⌉, right ⌊n/2⌋ bits of y
    P1, P2, P4 = MUL(xL, yL), MUL(xL+xR, yL+yR), MUL(xR, yR)
    return  P1 2^n + (P2-P1-P4) 2^(n/2) + P4
```

$a = 3 \quad b = 2$

$d = 1$

$d = 1 < \log_2 3$

$O(n^{\log_2 3})$

$$T(n) = 3T\left(\frac{n}{2}\right) + O(n) \rightarrow O(n^{\log_2 3})$$

$\hookrightarrow$ or $O(n^{1.59})$

**Facts** ① for a geometric progression $1, \alpha, \alpha^2 \dots \alpha^k$ with ratio $\alpha$ we have:

$$1 + \alpha + \alpha^2 \; \dots \; \alpha^k \longrightarrow \begin{cases} O(1) & \alpha < 1 \\ O(k) & \alpha = 1 \\ O(\alpha^k) & \alpha > 1 \end{cases}$$

② $$2^{\log_2 n} = n \qquad\qquad 4^{\log_2 n} = \left(2^{\log_2 n}\right)^2 = n^2$$

③ $$a^{\log b} = b^{\log a}$$

$$\left(b^{\log_b a}\right)^{\log b}$$

---

**Recurrence theorem** (Master's theorem)

for $a > 0$, $b > 1$, $d \geq 0$ if $T(n) = a\,T\left(\left\lceil \frac{n}{b} \right\rceil\right) + n^d$

then

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

| | Size of the problem | # of instances |
|---|---|---|
| Level-0 | $n$ | $1$ |
| Level-1 | $\frac{n}{b} \; \frac{n}{b} \dots \frac{n}{b}$ | $a$ |
| Level-2 | $\frac{n}{b^2} \; \frac{n}{b^2} \; \frac{n}{b^2}$ | $a^2$ |
| $\vdots$ | $\vdots$ | |
| Level-$\log_b n$ | $1 \; 1 \; 1 \; 1 \dots 1$ | $a^{\log_b n}$ |

$$1 \times c n^d + a \times c\left(\frac{n}{b}\right)^d + a^2 \times c\left(\frac{n}{b^2}\right)^d \dots\dots \; a^{\log_b n} \; c\left(\frac{n}{b^{\log_b n}}\right)^d$$

$$= c n^d \times \left[1 + \frac{a}{b^d} + \left(\frac{a}{b^d}\right)^2 \dots\dots \left(\frac{a}{b^d}\right)^{\log_b n}\right]$$

Case ① $\frac{a}{b^d} < 1 \iff d > \log_b a \longrightarrow O(n^d)$

Case ② $\frac{a}{b^d} = 1 \iff d = \log_b a \longrightarrow O(n^d \log n)$

Case ③ $\frac{a}{b^d} > 1 \iff d < \log_b a \longrightarrow O\left(\cancel{n^d} \times \frac{a^{\log_b n}}{\left(b^{\log_b n}\right)^{\cancel{d}}}\right)$

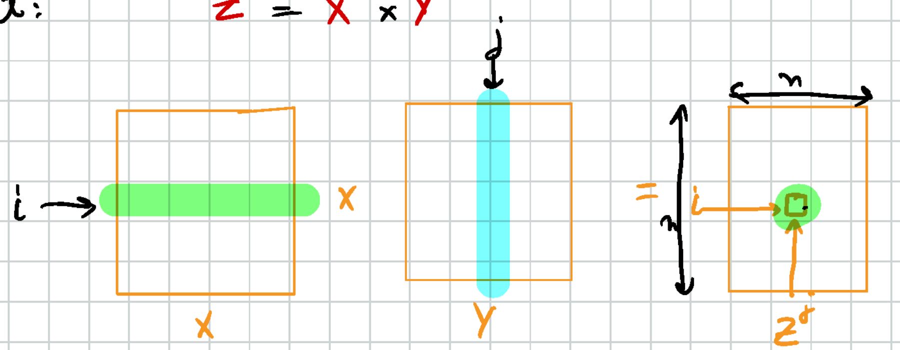
$$\downarrow$$

$$O\left(n^{\log_b a}\right)$$

## Matrix multiplication

Input: two $n \times n$ matrices $X, Y$

Output: $Z = X \times Y$




$O(n)$

① $n$ multiplications

② sum of the results

↳ $O(n)$

$O(n)$

$$n^2 \times O(n) = O(n^3)$$

1969 Strassen

$$X = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \qquad Y = \begin{bmatrix} E & F \\ G & H \end{bmatrix}$$

$A, B, C \dots H$ are now $\frac{n}{2} \times \frac{n}{2}$ matrices

$$XY = \begin{bmatrix} \boxed{A}\boxed{E} + BG & AF + BH \\ CE + DG & CF + DH \end{bmatrix}$$

$$T(n) = \underset{a=8}{8}\, T\Big(\frac{n}{\underset{b=2}{2}}\Big) + O(n^2) \qquad d=2$$

$2 = d < \log_b a = 3$

$$\downarrow$$

$$O(n^3)$$

$$X \times Y = \begin{bmatrix} P_5 + P_4 - P_2 + P_6 & P_1 + P_2 \\ P_3 + P_4 & P_1 + P_5 - P_3 - P_7 \end{bmatrix}$$

$P_1 = A(F - H)$

$P_2 = (A + B)H$

$P_3 = (C + D)E$

$P_4 = D(G - E)$

$P_5 = (A+D)(E+H)$

$P_6 = (B-D)(G+H)$

$P_7 = (A - C)(E + F)$

$$T(n) = \underset{a}{7}\, T\Big(\frac{n}{\underset{b}{2}}\Big) + O(n^2) \qquad d=2$$

$2 = d < \log_b a = \log_2 7$

$$\downarrow$$

$$= O(n^{\log_2 7}) = O(n^{2.81})$$

# Median Finding

Input: A list of $n$ numbers (unsorted)

Output: Find the $\lceil \frac{n}{2} \rceil^{th}$ smallest number in the list

Example: $[45, 1, 10, 30, 25]$

Find the $50^{th}$ percentile number in this list?

$[\underbrace{1 \cdots\cdots 1}_{100}]$ → mean, median = 1

$[\underbrace{1 \cdots 1}_{99}\ 10{,}000]$

mean > 100, median = 1

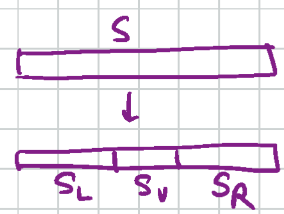
Naive: ① Sort the list → $O(n \log n)$

② pick the $\lceil \frac{n}{2} \rceil$ number from the list → $O(1)$

# Select

Input: unsorted list $S = \{a_1 \ldots a_n\}$, integer $k \in \{1 \ldots n\}$

Output: $k^{th}$ smallest number in $S$




* Select $(S, 1)$ → smallest number in $S$

* Median $(S)$ → Select$(S, \lceil \frac{n}{2} \rceil)$

Algorithm:

* Pivot $v \leftarrow S = \{a_1 \ldots a_n\}$
* Split $S \rightarrow S_L \quad S_v \quad S_R$

$$S_L = \{a \in S \mid a < v\}$$

$$S_v = \{a \in S \mid a = v\}$$

$$S_R = \{a \in S \mid a > v\}$$

* case analysis

$k \le |S_L| \to \text{Select}(S_L, k)$

$|S_L| < k \le |S_L| + |S_v| \to \text{Return } v$

$|S_L| + |S_v| < k \to \text{Select}(S_R, k - |S_L| - |S_v|)$

Case I : (Badcase) pivot was the smallest or the largest element in S

$$O(n) + O(n-1) \ldots\ldots O(1) \to O(n^2)$$

Case II : (Ultra Good) pivot $v = k^{th}$ smallest element.

Good $\quad |S_L| \approx |S_R| \approx \frac{n}{2}$

$$O(n) + O\left(\frac{n}{2}\right) + \ldots \; O\left(\frac{n}{2^{\log n}}\right)$$

$$= \underline{O(n)}$$