divide and conquer

$\boxed{A_L \mid A_R} \times \boxed{B_L \mid B_R}$

A $\qquad$ B

n bits $\qquad$ n bits

$A = \boxed{A_L \mid 0000} + \boxed{0000 \mid A_R}$

$101111 = 101000 + 000111$

$$A = A_L \cdot 2^{\frac{n}{2}} + A_R.$$

$$B = B_L \cdot 2^{\frac{n}{2}} + B_R,$$

$$\underbrace{A \times B}_{n \text{ bit}} = 2^{n} \underbrace{A_L \cdot B_L}_{\frac{n}{2} \text{ bit}} + 2^{\frac{n}{2}} \underbrace{(A_L B_R + A_R B_L)}_{\frac{n}{2}} + \underbrace{A_R B_R}_{\frac{n}{2}}.$$

☆ MULTIPLY (A, B)

$A = 2^{\frac{n}{2}} A_L + A_R \qquad B = 2^{\frac{n}{2}} B_L + B_R.$

$P_1 \leftarrow$ MULTIPLY (A_L, B_L)

$P_2 \leftarrow$ M (A_L, B_R)

$P_3 \leftarrow$ M (A_R, B_L)

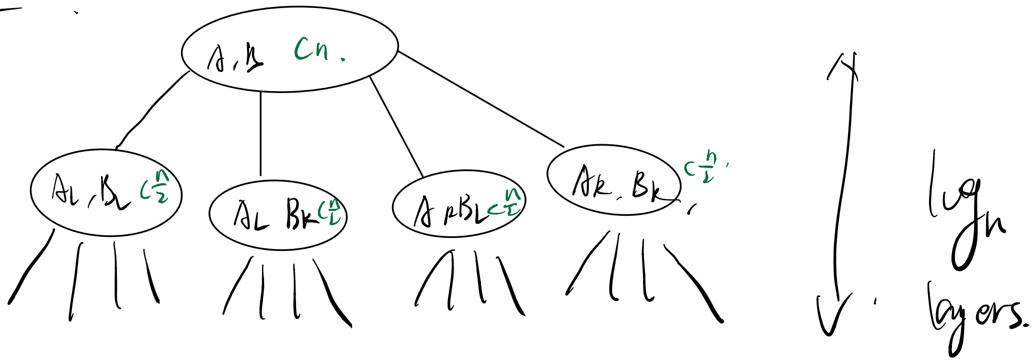
$P_4 \leftarrow$ M (A_R, B_R)

Base Case,

$n = 1.$

Return: $2^{n} \cdot P_1 + 2^{\frac{n}{2}} (P_2 + P_3) + P_4.$

EXECUTION



Return: $2^n \cdot P_1 + 2^{\frac{n}{2}}(P_2+P_3) + P_4$.

$O(n)$ steps $=$ $c \cdot n$ steps

$$1 \cdot cn + 4 \cdot \frac{cn}{2} + 4^2 \frac{cn}{2^2} + \cdots + 4^{\log n} \frac{cn}{2^{\log n}}.$$

$$= cn + cn\frac{4}{2} + cn\frac{4^2}{2^2} + \cdots + cn\left(\frac{4}{2}\right)^{\log n}.$$

FACT In a GP (geometric progression) sum $\approx O$(last term).

$a^{\log b} = b^{\log a}$.

$= O(n^2)$.

Better Way:

cut children from $4 \to 3$.

Then $t = O(n^{\log 3})$

How ?

$P_1 = (A_L + A_R)(B_L + B_R)$

$P_2 = A_L B_L$

$P_3 = A_R B_R$.

$$A \times B = 2^n \underbrace{A_L \cdot B_L}_{P_2} + 2^{\frac{n}{2}} \underbrace{(A_L B_R + A_R B_L)}_{P_1 - P_2 - P_3} + \underbrace{A_R B_R}_{P_3}.$$

☆ MULTIPLY (A, B)

$A = 2^{\frac{n}{2}} A_L + A_R \qquad B = 2^{\frac{n}{2}} B_L + B_R$.

$P_1 \leftarrow$ MULTIPLY $(A_L + A_R, B_L + B_R)$

$P_2 \leftarrow$ M (A_L, B_L)

$P_3 \leftarrow$ M (A_R, B_R)

Base case, $n = 1$.

Return: $2^n \cdot P_2 + 2^{\frac{n}{2}} (P_1 - P_2 - P_3) + P_3$

$$\boxed{\begin{aligned} T[n] &= 3 \cdot T[\tfrac{n}{2}] + cn \\ T[1] &= 1 \end{aligned}}$$

for example. $T[n] = T[n-1] + n$

$$= T[n-2] + n-1 + n$$

$$= T[n-3] + n-2 + n-1 + 1.$$

$$= T[n-k] + (n-k+1) + (n-k+2) \dots + n.$$

$$= T[1] + 2 + \dots + n$$

$$= 1 + 2 + 3 \dots + n. = \frac{(n+1)n}{2}$$

$$T[n] = 3T[n/2] + c\cdot n.$$

$$= 3\left[3\cdot T[n/2\cdot 2] + c\cdot \frac{n}{2}\right] + c\cdot n.$$

$$= 3^2\cdot T[n/2^2] + cn\cdot (\tfrac{3}{2}) + cn$$

$$= 3^2\left[3\cdot T[n/2^2\cdot 2] + cn/2^2\right] + cn(\tfrac{3}{2}) + cn.$$

$$= 3^3 T[n/2^3] + cn(\tfrac{3}{2})^2 + cn(\tfrac{3}{2}) + cn.$$

$$= 3^k T[n/2^k] + cn[1 + \tfrac{3}{2} + \dots + (\tfrac{3}{2})^{k-1}]$$

$2^{\log n} = n.$

$k = \log n.$

$$= 3^{\lg n}\cdot T[n/2^{\lg n}] + cn[1 + \tfrac{3}{2} + \dots + (\tfrac{3}{2})^{\log n - 1}]$$

$$= n^{\log 3} + O(cn\,(\tfrac{3}{2})^{\lg n - 1})$$

$$= n^{\log 3} + O(cn\cdot \frac{3^{\lg n - 1}}{2^{\log n - 1}})$$

$$= n^{\log 3} + O(n\cdot \frac{3^{\log n}/3}{2^{\log n}/2})$$

$$= n^{\log 3} + O(n^{\log 3})$$

LET $a, b, c \in R^+$ $b > 1$.

$$\boxed{T[n] = a\cdot T[n/b] + O(n^c) \qquad T[1] = 1}$$

Case 1: $c < \log_b a$ $O(n^{\log_b a})$

Case 2: $c = \log_b a$ $O(n^c \log n)$

Case 3: $c > \log_b a$ $O(n^c)$.

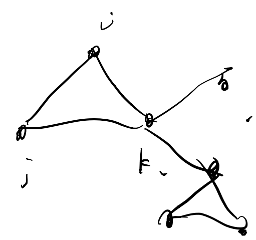
MATRIX MULTIPLICATION

Number are 32 bit.

TRIANGLE.

Network (G, V, E)

Is there a $\triangle$?



Inner product: $u = (u_1, \dots, u_n)$, $v = (v_1, \dots, v_n)$ real vectors

$$\boxed{\langle u, v \rangle = \sum_{i=1}^{n} u_i v_i}$$

$O(n)$ time.

$$\underbrace{X}_{n} \cdot \underbrace{Y}_{n} = \underbrace{Z}_{n}$$

$O(n^3)$ time.

$Z_{ij} = \langle i^{th}$ row of X, j^{th} column of $Y \rangle$.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}$$

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

Recursively compute $\{\overset{P_1}{AE}, \overset{P_2}{BG}, \overset{P_3}{AF}, \overset{P_4}{BH}, \overset{P_5}{CE}, \overset{P_6}{DG}, \overset{P_7}{CF}, \overset{P_8}{DH}\}$

Return. $\begin{bmatrix} P_1+P_2 & P_3+P_4 \\ P_5+P_6 & P_7+P_8 \end{bmatrix}$

$$T(n) = 8 \cdot T(n/2) + O(n^2)$$

$a=8$

$b=2$ $\Rightarrow$ $O(n^3)$

$c=2$

Better Way?

$P_1 = A \cdot (F-H)$

$P_2 = (A+B) \cdot H$

$P_3 = (C+D) \cdot E$

$P_4 = D(G-E)$

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

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

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

Return $\begin{bmatrix} P_5+P_4-P_2+P_6 & P_1+P_2 \\ P_3+P_4 & P_1+P_4-P_3-P_7 \end{bmatrix}$

$$= \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}$$

$$T(n) = 7\,T(n/2) + O(n^2)$$

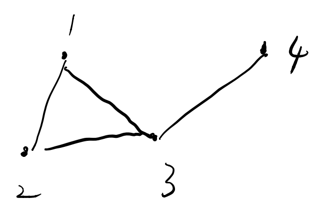
$$T(n) = n^{\log_2 7} \simeq n^{2.7}$$

Back to triangle detection problem.

k.

Given an unweighted graph G. Adjacency matrix, entries in $\{0, 1\}$

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$



① Compute Boolean Product.

$$C = A \times A$$

$$C_{ij} = \bigvee_k A_{ik} \wedge A_{kj}.$$

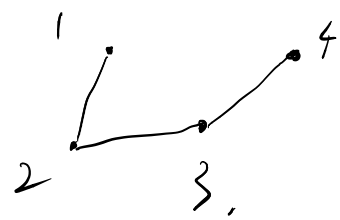
$$C = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 3 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

② . compute $\bigvee_{ij} A_{ij} \wedge C_{ij}$

this equals $\bigvee_{ijk} A_{ij} \wedge A_{ik} \wedge A_{kj}$

$$ans = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \text{exist .}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



$$C = A \times A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

$$\bigvee_{ij} A_{ij} \wedge C_{ij} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$\downarrow$

not exist

MEDIAN

$\text{Median}(\{a_1, \cdots a_n\}) = \lceil \frac{n}{2} \rceil^{th}$ smallest number in the list.

$\Downarrow$ more generally.

SELECT $(\{a_1, \cdots a_n\}, k)$: find the kth smallest.

① Pick an element $v = a_i$ (pivot)
random

$S_< = \{a_i \mid a_i < v\}$ $S_= = \{a_i \mid a_i = v\}$ $S_> = \{a_i \mid a_i > v\}$

5, 8, 9, 9, 9, 10, 11, 13, 14, 15, 16, 17, 20.

pivot = 9.

Case: $k \le |S_<|$ Return SELECT $(S_<, k)$.

Case: $|S_<| < k \le |S_<| + |S_=|$ Return v.

Case: $|S_<| + |S_=| < k$. Return SELECT $(S_>, k - |S_<| - |S_=|)$