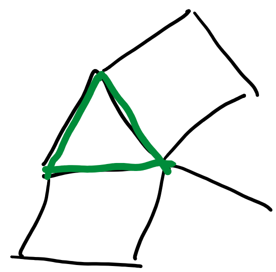
22 Triangle Detection & Matrix Multiplication

Tuesday, April 8, 2025 13:56

Triangle Detection: Given graph $G=(V,E)$
return Yes/No if there is a triangle in the graph
(clique of size 3)



Brute Force $O(n^3)$

$\binom{n}{3} \le n^3 \qquad \binom{n}{k} \sim n^k$

TriangleDetection$(G=(V,E))$

$A =$ adjacency matrix // $A_{ij} = \begin{cases} 1 & \text{if } \{i,j\} \in E \\ 0 & \text{else} \end{cases}$

$M = A \cdot A$

for $\{u,v\} \in E$
    if $M_{u,v} \neq 0$
        return true
return false

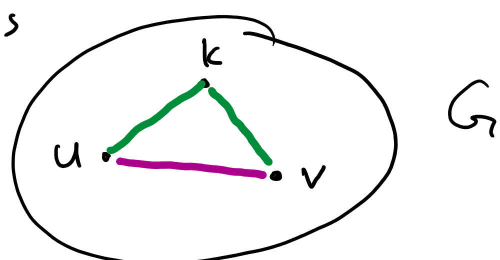
$$M_{uv} = (A \cdot A)_{uv} = \sum_k A_{uk} \cdot A_{kv} \neq 0$$

$\iff \exists k \quad A_{uk} = 1 \;\&\; A_{kv} = 1$

$\iff \exists k \quad \{u,k\} \in E \quad \{k,v\} \in E$

$(A^2)_{uv} \neq 0 \iff \exists$ path of length 2 from u to v

$\Rightarrow$ If we can multiply $n \times n$ matrices in time $T(n)$
then we can detect triangles in time $O(T(n) + n^2)$
$= O(T(n))$ because $T(n) \ge n^2$ (need to read $n \times n$ input matrix)



Benefit in practice: There are libraries that handle hardware / low level optimization of matrix mult.
$\Rightarrow$ we can directly use that without knowing those details

$\Rightarrow$ we can directly use that without knowing those details

Benefit in theory: $n \times n$ matrices can be multiplied in $O(n^{2.372})$ time [Alman Williams 22]

$\Rightarrow$ detect triangles in $O(n^{2.372})$ time.

Theorem: If we can detect triangles in subcubic time $O(n^{3-\varepsilon})$ then we can multiply (boolean) matrices in subcubic time.

"BMM Conjecture": There is no "efficient" subcubic time algorithm for (boolean) matrix multiplication

$\uparrow$ boolean matrix multiplication

Def: Boolean Matrix Multiplication

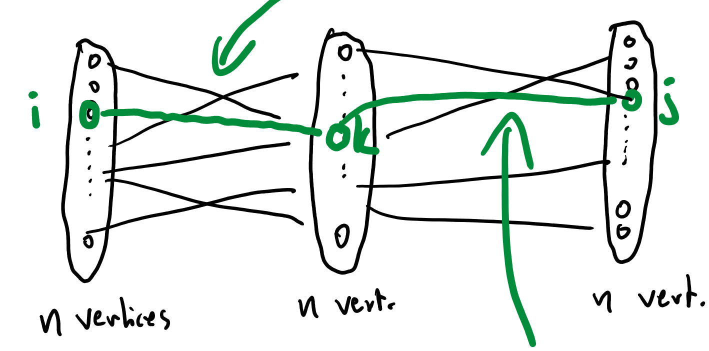
Given $A, B \in \{0,1\}^{n \times n}$ return $C \in \{0,1\}^{n \times n}$

$C_{ij} \neq 0 \iff (A \cdot B)_{ij} \neq 0$

example $C_{ij} = \min(1, (A \cdot B)_{ij})$

$A, B \in \{0,1\}^{n \times n}$

$\{i,k\} \in E \iff A_{ik} = 1$



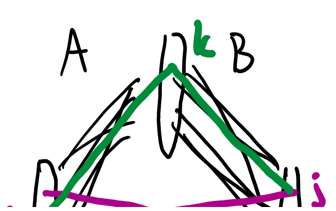
$\{k,j\} \in E \iff B_{kj} = 1$

$$(A \cdot B)_{ij} = \sum_k A_{ik} \cdot B_{kj}$$

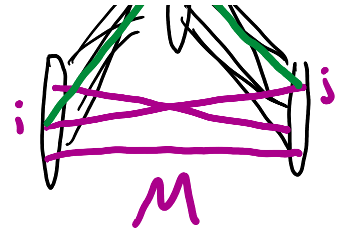
$$\neq 0$$

$\iff \exists k \;\; A_{ik} = 1 \;\&\; B_{kj} = 1$

$\iff \exists$ path $i \to k \to j$ length 2



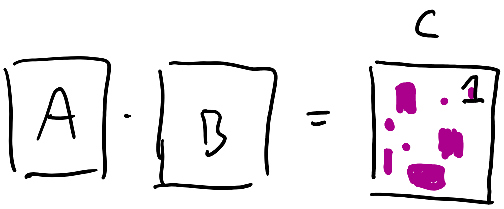
$M \in (\{1 \ldots n\} \times \{1 \ldots n\})$



$M \subseteq (\{1 \dots n\} \times \{1 \dots n\})$

$(i,j) \in M$

G has a triangle $\iff \exists \, \underline{(i,j) \in M}$ where $\underline{(A \cdot B)_{ij} \neq 0}$



Mask M, entries we care about

Lemma: Assume we can detect Δ in time $T(n)$
then given $A, B \in \{0,1\}^{n \times n}$ and a mask $M \subseteq \{1 \dots n\}^2$
we can <u>detect</u> if $\exists \, (i,j) \in M$ where $(A \cdot B)_{ij} \neq 0$

1) Corollary: Given A, B mask M we can <u>find</u> <u>one</u> $(i,j) \in M$
where $(AB)_{ij} \neq 0$. in time $O(T(n) \log n)$ (Do binary search on M)

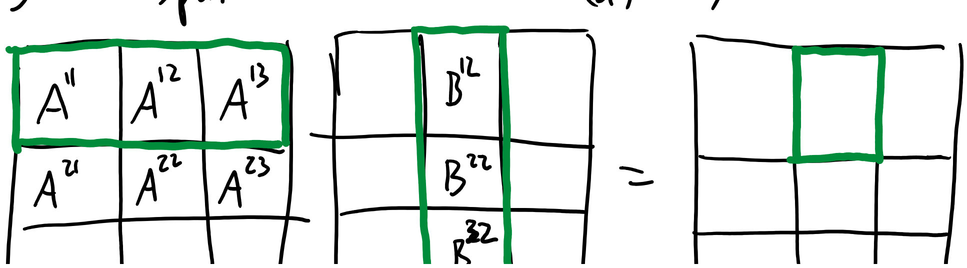
2) Given A, B, mask M we can find all $(i,j) \in M$
where $(AB)_{ij} \neq 0$ in time $O(T(n) \log n \cdot (1 + k))$

$k = \# (i,j) \in M$ where $(AB)_{ij} \neq 0$

$\Rightarrow$ can compute AB in time $O(T(n) \cdot \log(n) \cdot n^2)$

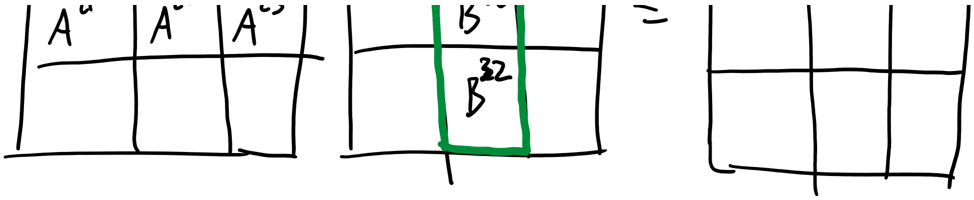
$k \leq n^2$

2:53 Split $n \times n$ matrices into $\left(\frac{n}{d}\right)^2$ many $d \times d$ blocks

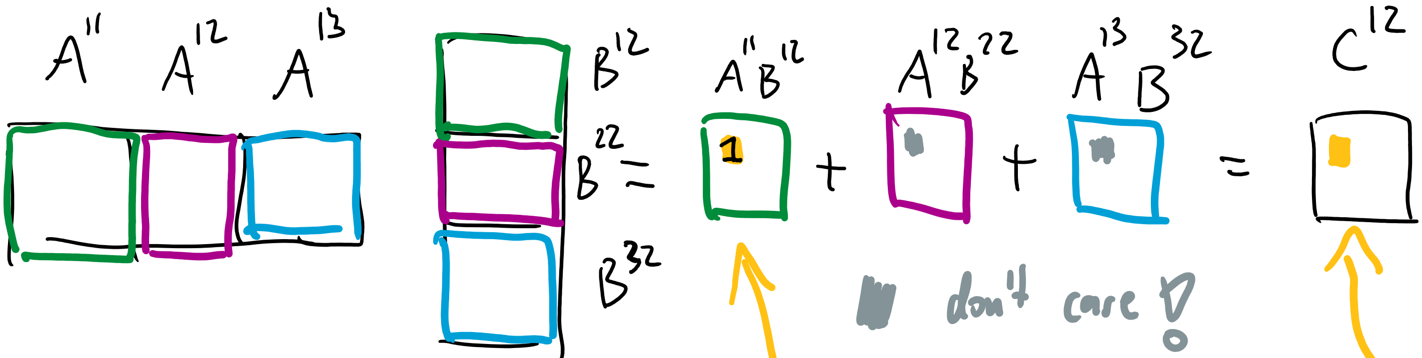


$$C^{ij} = \sum_{k=1}^{n/d} A^{ik} \cdot B^{kj}$$

$C^{12} = A^{11} B^{12} + A^{12} B^{22}$



$$C^{12} = A^{11}B^{12} + A^{12}B^{22} + A^{13}B^{32}$$



If yellow entry is non-zero then we do not care about the gray entries, because we already know entry is non-zero.

BMM(A, B)

Split A and B into $\frac{n}{d} \times \frac{n}{d}$ many $d \times d$ matrices

for $i = 1 \ldots n/d$
  for $j = 1 \ldots n/d$
    $C^{ij} = 0$ matrix
    $M = \{1 \ldots d\}^2$
    for $k = 1 \ldots n/d$
      T = find all positions $(a,b) \in M$ with $(A^{ik} \cdot B^{kj})_{a,b} \neq 0$
      $C_{ab} = 1$ for all $(a,b) \in T$
      $M = M \setminus T$

combine all C^{ij} into $n \times n$ matrix C

return C

Complexity: Consider $\overbrace{\text{blue}}^{\text{one iteration of}}$ loop. Each $(a,b) \in \{1..d\}^2$ is returned only <u>once</u> as a nonzero entry

$\Rightarrow$ total # of nonzero entries we detect within blue part is bounded by d^2

$\Rightarrow$ one iteration of blue loop is

$$O\left(T(d)\log(n)\cdot\left(\frac{n}{d}+d^2\right)\right)$$

$\Rightarrow$ total time $O\left(T(d)\log(n)\left(\frac{n}{d}+d^2\right)\cdot\left(\frac{n}{d}\right)^2\right)$

$$d^{3-\varepsilon}\log(n)\cdot\left(\frac{n}{d}+d^2\right)\cdot\left(\frac{n}{d}\right)^2 = n^2\cdot d^{1-\varepsilon}\left(\frac{n}{d}+d^2\right)\log(n)$$

$$(d=n^{1/3}) = n^2\, n^{1/3-\varepsilon/3}\, n^{2/3}\log(n)$$

$$= n^{3-\frac{\varepsilon}{3}}\log(n)$$

If we can detect Δ in $O(n^{3-\varepsilon})$

then we can compute $A\cdot B$ in $O\left(n^{3-\frac{\varepsilon}{3}}\right)$