Strassen '69 Matrix Multiplication

Key Insight: Use fewer (7 instead of 8) multiplications to compute $A \times B$ when $A, B \in \mathbb{R}^{2\times 2}$

Naive

A B C

$a_{11}\ a_{12}$ $b_{11}\ b_{12}$ $c_{11}\ c_{12}$

$a_{21}\ a_{22}$ $b_{21}\ b_{22}$ $c_{21}\ c_{22}$

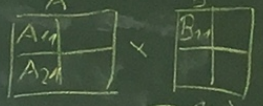
$c_{11} = a_{11} \cdot b_{11} + a_{12} \cdot b_{21}$ 8 multiplications

$c_{12} = a_{11} \cdot b_{12} \dots$

$c_{21} =$;

$c_{22} =$

Recursive Scheme



$T(n) \leq 8\,T(\frac{n}{2}) + O(n^2)$

$\Rightarrow T(n) \leq O(n^3)$

Strassen's Improvement:

$m_1 = (a_{11} + a_{22}) \cdot (b_{11} + b_{22})$

$m_2 = (a_{21} + a_{22}) \cdot b_{11}$

$m_3 = a_{11} \cdot (b_{12} - b_{22})$

$m_4 = a_{22} \cdot (b_{21} - b_{11})$

$m_5 = (a_{11} + a_{12}) \cdot b_{22}$

$m_6 = (a_{21} - a_{11}) \cdot (b_{11} + b_{12})$

$m_7 = (a_{12} - a_{22}) \cdot (b_{21} + b_{22})$

$c_{11} = m_1 + m_4 - m_5 + m_7$

$c_{12} = m_3 + m_5$

$c_{21} = m_2 + m_4$

$c_{22} = m_1 - m_2 + m_3 + m_6$

7 mult.

$T(n) \leq 7\,T(\frac{n}{2}) + O(n^2)$

$\Rightarrow T(n) \leq O(n^{\log_2 7}) = O(n^{2.8074\dots})$

Remark: Alg works for any ring (eg. $(\mathbb{R}, +, \cdot)$) but not for any semi-ring (e.g. $(\mathbb{R}, \min, +)$)

Reason: Alg. exploits inverse element w.r.t addition

This is what makes alg. "non-combinatorial"

E.g. BMM $(\{0,1\}, \vee, \wedge) \to (\mathbb{R}, +, \cdot)$

Boolean Matrix Multiplication

Def: (BMM)

Input: Boolean Matr. $A, B \in \{0,1\}^{n \times n}$

Task: Compute C

$$\forall i,j \quad C[i,j] = \bigvee_{k=1}^{n} A_{ik} \wedge B_{kj}$$

$$= \begin{cases} 1 & \text{if } \exists k \text{ s.t. } A[i,k]=1 \text{ and } B[k,j]=1 \\ 0 & \end{cases}$$

Matrix product in Boolean semiring

Def: ω is the infimum over all γ s.t. matrix mult. has $O(n^\gamma)$-time alg

$\Rightarrow$ MM is in time $O(n^{\omega+\varepsilon})$ for any $\varepsilon > 0$

"Sloppy": MM is in time $O(n^\omega)$

Def: alg is combinatorial if it does not rely on an $O(n^\omega)$-time alg. for MM

Open Question:

$\omega = 2$?

Algorithms:

- $O(n^3)$
- $O(n^3/\log^2 n)$ [Arlazarov et. al '70] "4 Russians" 4R "combinatorial"
- $O\left(\frac{n^3 \cdot \text{poly}(\log\log n)}{\log^4 n}\right)$ [Yu '15]
- Reduction to MM on real numbers $\Rightarrow O(n^\omega)$

$$C[i,j] = \bigvee_{k=1}^{n} A[i,k] \wedge B[k,j]$$

vs. $\hat{C}[i,j] = \sum_{k=1}^{n} A[i,k] \cdot B[k,j]$ "inner product"

Obs: $C[i,j] = 0$ iff $\hat{C}[i,j] = 0$

- $O(n^{2.81})$ [Strassen '69]
- $O(n^{2.38})$ [Coppersmith/Winograd '87]
- $O(n^{2.37})$ [LeGall '14]

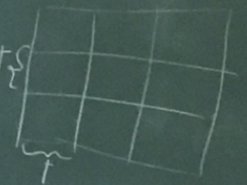
"non-combinatorial"

4 Soviets: $O(n^3/\log n)$ $C = A \times B$ (Boolean)

Idea ① Preprocess A and construct lookup table

② Speed up naive alg using lookup table

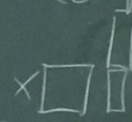
① Divide A into blocks of size $t \times t$ (where $t = 0.5 \log n$)



$\Rightarrow$ blocks $= (\frac{n}{t})^2 = O(\frac{n^2}{\log^2 n})$

For each of the blocks X

- Precompute $X \cdot v$ for every vector $v \in \{0,1\}^t$

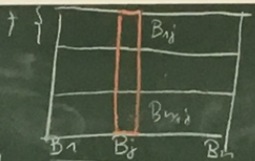


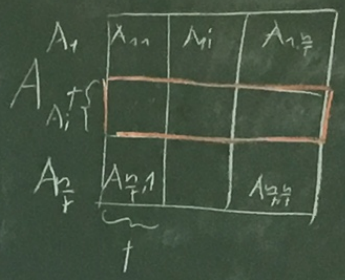
t inner products: time $O(t^2)$ per X and v

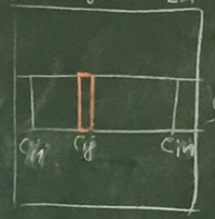
- Store each result $X \cdot v$ in lookup table

Given X, v, retrieve $X \cdot v$ in time $O(t)$

Total prepr. time: $O((\frac{n}{t})^2 \cdot 2^t \cdot t^2) = O(n^2 \cdot 2^t) = O(n^{2.5})$ ($2^{0.5 \log n} = (2^{\log n})^{0.5}$)

② 





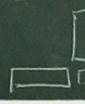
Compute vector $C_j = A_i \times B_j = A_{i1} \times B_{1j} \vee \dots \vee A_{i\frac{n}{t}} \times B_{\frac{n}{t}j}$

$= \bigvee_{k=1}^{n/t} \underbrace{A_{ik} \times B_{kj}}_{\text{Time } O(t) \text{ Tablelookup}}$

$\vee$ in each component of the vectors

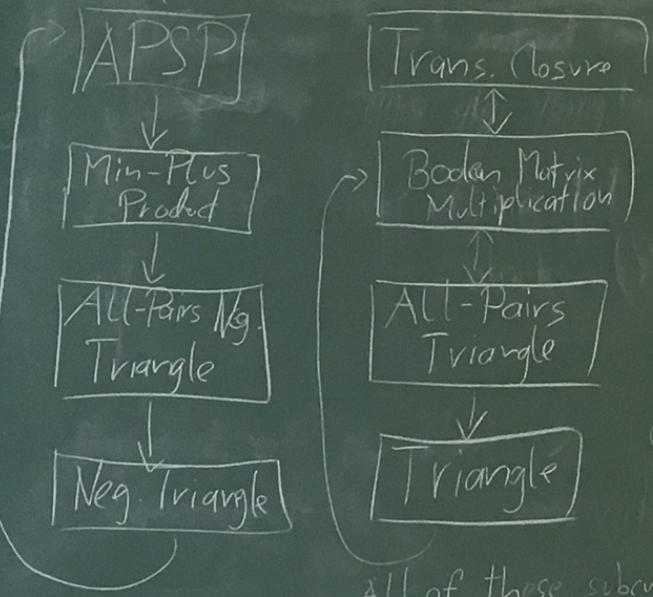
$\Rightarrow$ Compute C in time $O(n \cdot \frac{n}{t} \cdot \frac{n}{t} \cdot t) = O(\frac{n^3}{t}) = O(\frac{n^3}{\log n})$

Remark: On word RAM with word size $w = \Omega(\log n)$ we can exploit "bit-level" parallelism to shave another $\log n$



$\rightarrow O(1)$ time

Subcubic Equivalences (for "Combinatorial" Algorithms)



Given directed G, compute $\forall i,j \in V$ if there is a path from i to j in G
→ TC matrix

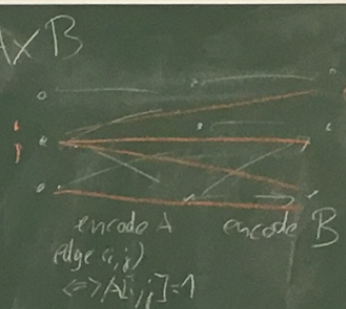
Given unwtd., undir. G, with $V = I \cup J \cup K$
Decide $\forall i \in I$, $\forall j \in J$ (triangle-free) $\exists k \in K$ s.t. (i,j,k) is a Δ?

Given unweighted, undir. G,
Does it contain a triangle?

All of these subcubic reductions are "combinatorial"

⇒ If one of these problems admits a truly subcubic alg, then all of them do ($O(n^{3-\varepsilon})$-time)

$A \times B$



BMM → All-Pairs Δ

Triangle → BMM

Let A be adj. matrix of G

$A[i,j] = 1$ iff $\exists$ edge (i,j)

1. Compute $C := A \times A$ — 1 oracle call

$$\forall i,j \quad C_{ij} = \bigvee_k A[i,k] \wedge A[k,j]$$

2. Compute $\bigvee_{i,j} A_{ij} \wedge C_{ij}$ — $O(n^2)$

$$= \bigvee_{i,j,k} A_{ij} \wedge A_{ik} \wedge A_{kj}$$

= 1 iff G has a triangle

$O(n^2 d) = O(n^3)$
$= O(n^{2.81})$?