APSP: given L ($L_{ij} = \ell(ij)$), compute D ($D_{ij} = d(i,j)$) $\Theta(n^3)$

Transitive Closure: given A (adj. matrix), compute A^* (reachability mat.)

Matrix Multiplication: $O(n^3)$ by def.

in a ring

$O(n^{2.808})$ Strassen 1969

$O(n^{2.376})$ Coppersmith & Winograd 1990

$O(n^{2.373})$ Williams 2012

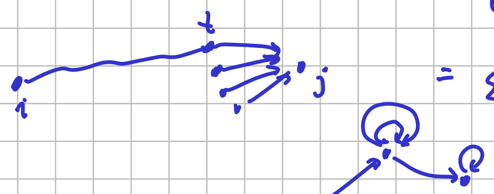
$O(n^{2+\varepsilon})$ conjecture

$\Omega(n^2 \log n)$ lower bound (restricted)

$O(n^\omega) \leftarrow$ }

A^k_{ij} = # ij-walks with exactly k edges (by induction) $(M \cdot A)_{ij} = \sum_t M_{it} \underbrace{A_{tj}}$

$(A+E)^k_{ij} > 0 \iff \exists$ ij-walk with at most k edges $= \sum$ $[tj \in E]$



$(A+E)^n > 0 \iff A^*_{ij} = 1$

$\underbrace{((A^2)^2)^2 \ldots}_{k} = A^{2^k}$ … $k := \lceil \log n \rceil$ in k mat. multiplications we compute A^{2n}

$O(n^\omega \cdot \log n)$

↳ & replace non-zeroes by ones after every multiplication

→ all entries $\leq n$

$A^{2k} = (A^k)^2$

$A^{2k+1} = (A^k)^2 \cdot A$

} $O(\log n)$ steps each in $O(n^\omega)$ time → total $O(n^\omega \cdot \log n)$

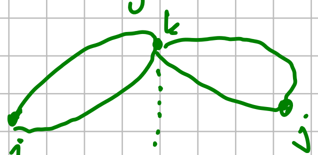
Algebra $(X, \oplus, \otimes)$

Df: $(\oplus,\otimes)$-product of matrices in $X^{n\times n}$: $(A \cdot B)_{ij} := \bigoplus_k A_{ik} \otimes B_{kj}$.

std. mat. mult. $(+,\cdot)$-product

$(\cup,\cdot)$-product (matrices of bundles) A_{ij} B_{ij} ij-bundles

$\bigcup_k A_{ik} \cdot B_{kj}$



if $A_{ij} = e_{ij}$ or $\emptyset$, $A_{ii} = \varepsilon_i$

A^t_{ij} = bundle of all ij-walks with exactly t edges

Walk algebra:

$\cup$ union

$\cdot$ concatenation

u v w

$*$ iteration

constants

f: bundles → values

$f(A \cup B) \leftarrow f(A), f(B)$

$(\vee,\wedge)$-products $(A \vee E)^n = A^*$

↳ can be computed via std. mat. mult. in $O(n^\omega)$

↳ reachability can be computed in $O(\log n)$ $(\vee,\wedge)$-products → $O(n^\omega \log n)$

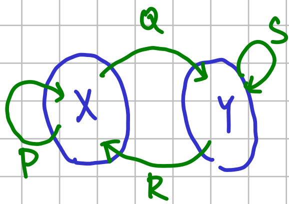
$(\min,+)$-products $L^n = D$ APSP in $O(\log n)$ $(\min,+)$-products

↳ $O(n^3)$ by def.

$O(n^3/\log n)$ [Chan 2008] → $O(n^3)$

more efficient algs for small integers

Divide & Conquer for reachability using $(\vee,\wedge)$-products



$$A = \begin{array}{c|cc} & X & Y \\ \hline X & P & Q \\ Y & R & S \end{array} \longrightarrow A^* = \begin{array}{c|cc} & X & Y \\ \hline X & I & J \\ Y & K & L \end{array}$$

$I = (P \vee QS^*R)^*$

$J = IQS^*$

$K = S^*RI$

$L = S^* \vee S^*RIQS^*$

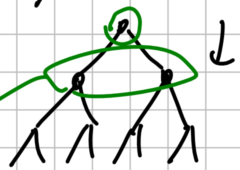
2 recursive calls for size $n/2$

$O(1)$ $(\vee,\wedge)$-products $\mu(n) \in \Omega(n^2)$

$O(1)$ cheap matrix operations $O(n^2)$

$$T(n) = 2T(n/2) + \underset{O(\mu(n))}{\Theta(\mu(\tfrac{n}{2}))}$$

$$T(n) = \Theta(\mu(n))$$



problems $\cdot 2$

size $/2$

work per problem decreases at least 4 times

work per level decreases at least 2 times

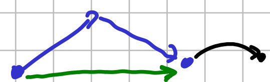
→ $O(n^\omega)$ for reachability

$(\min,+)$-products → $O(n^3/\log n)$ for APSP

Seidel's alg.

APSP in undirected unit-length graphs

WLOG G is connected



$G^2 = (V, E^2)$ $\quad ij \in E^2 \equiv \exists$ ij-walk in G with at most 2 edges

$A(G^2)$ can be computed from A by mat. mult.

$$\begin{tikzcd} G \arrow[r] & G^2 \\ A \arrow[r, "\text{squaring}"] & A' \arrow[d, "\text{APSP}"] \\ D & D' \arrow[l] \end{tikzcd}$$

• after $\log n$ squarings, G becomes complete → D is trivial

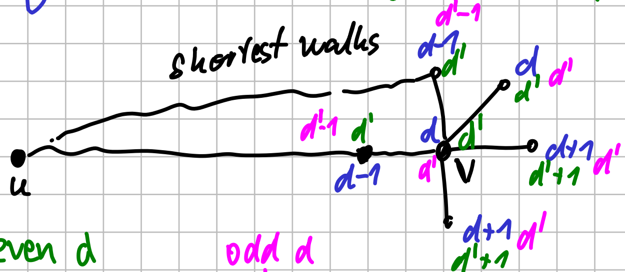
Fix u, $d(v) := D_{uv}$, $d'(v) := D'_{uv}$

Goal: given d', compute d

HW $\quad d'(v) = \lceil d(v)/2 \rceil$

$$d(v) = \begin{cases} 2d'(v) \\ 2d'(v) - 1 \end{cases}$$

it suffices to know the parity of $d(v)$



even d: all neighbors have $\geq d'$

odd d: all neighbors have $\leq d'$, at least one has $d'-1$

average / sum d' over neighbors of v

$\geq d' \cdot \deg(v)$ $\qquad < d' \cdot \deg(v)$

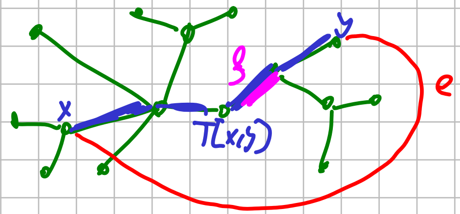
$$(D' \cdot A)_{ij} = \sum_k D'_{ik} \cdot \underbrace{A_{kj}} = \sum_{kj \in E} D'_{ik}$$

for $i = u$ this is sum of $d(k)$ over neighbors of j

$O(\log n)$ levels of recursion, each uses $2\times$ MM

→ $O(n^\omega \cdot \log n)$ total time.

Minimum Spanning Trees



- subgraph: $T \subseteq E$ (V is always the same)
- weights: $w: E \to \mathbb{R}$

  $w(T) := \sum_{e \in T} w(e)$
- MST: $T \subseteq E$ which is a tree s.t. $w(T)$ is min.

for disconnected graph: min. spanning forest

- for a tree T:
  - $T[x,y]$ = path in T between x,y
  - if $uv \in E \setminus T$: $T[uv] := T[u,v]$

    path covered by edge uv
- WMLOG we assume all weights distinct
- $e \in E \setminus T$ is <u>T-light</u> $\equiv \exists f \in T[e] : w(f) > w(e)$

(~~hcls~~)

<u>Thm:</u> T is a MST

$\Updownarrow$

there are no T-light edges

if T is a MST

$e \in E \setminus T$

$f \in T[e]$

$w(f) > w(e)$

e covers a <u>heavier</u> edge f

$T' := T + e - f$ is a spanning tree

<u>but</u> T' is lighter than T ↯