Timothy Chan

All Pairs Shortest Paths (APSP)

Dijkstra n times $\Rightarrow$ $O(n(n \log n + m))$

Floyd-Warshall $\Rightarrow$ $O(n^3)$

Subcubic?

Small integer weights from $\{1, \ldots, c\}$

undirected Chan '06 $O\left(\frac{mn}{\log n}(\log\log n)^2\right)$

Alon, Galil, Margalit '91 $O(n^{2.38})$

Seidel '92

directed Zwick '97 $O(n^{2.58})$

real vertex weights Chan '07 $O(n^{2.69})$

Real Euclidean weights in $\mathbb{R}^2$ Chan '07 $O(n^{2.93})$
(graph w/ arbitrary edges)

General real edge weights

→ today Fredman '76 $O\left(\frac{n^3}{\log^{1/3} n} (\log\log n)^{1/3}\right)$

Takaoka '91 $O\left(\frac{n^3}{\sqrt{\log n}} \sqrt{\log\log n}\right)$

Dobosiewicz '90 $O\left(\frac{n^3}{\sqrt{\log n}}\right)$ (independent)

Takaoka '04 $O\left(\frac{n^3}{\log n} \log\log n\right)$

Zwick '04 $O\left(\frac{n^3}{\log n} \sqrt{\log\log n}\right)$

→ Chan '05 $O\left(\frac{n^3}{\log n}\right)$

Han '06 $O\left(\frac{n^3}{\log^{5/4} n} (\log\log n)^{5/4}\right)$

Chan '07 $O\left(\frac{n^3}{\log^2 n} (\log\log n)^3\right)$

Han, Takaoka '12 $O\left(\frac{n^3}{\log^2 n} \log\log n\right)$

OPEN $O(n^{2.99})$ without matrix mult.

Williams '14 $O\left(\frac{n^3}{2^{c\sqrt{\log n}}}\right)$

uses matrix techniques & circuit complexity

→ Chan '15 $O\left(\frac{n^3}{\log^2 n} (\log\log n)^3\right)$ — same bound but simpler

Equivalent Problem. Min-Plus Matrix Multiplication
("Funny MM")

Given $n \times n$ matrices A, B

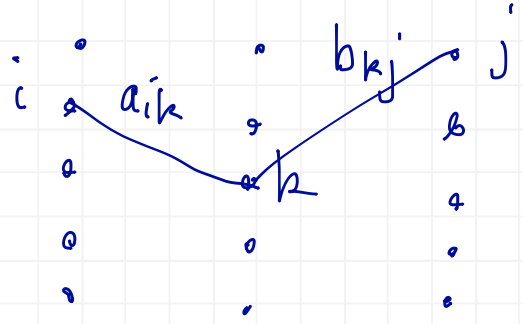
compute $c_{ij} = \min_{k=1..n} (a_{ik} + b_{kj})$

(can't use Strassen because don't have inverses)

Equivalence:

- reduce Funny MM to APSP

tripartite graph



- for other direction - repeated squaring. Costs $\log n$

Can eliminate the extra $\log n$ factor (Munro '71)

For rest of lecture: concentrate on Funny MM

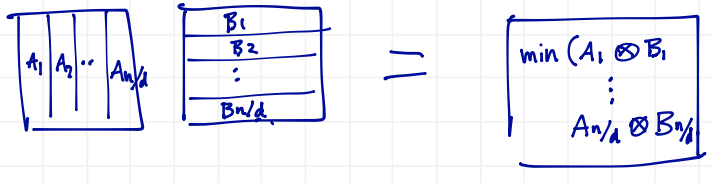
Fredman '76

Thm Funny MM can be solved in $O(n^{2.5} \log n)$ comparisons (but other operations more costly)

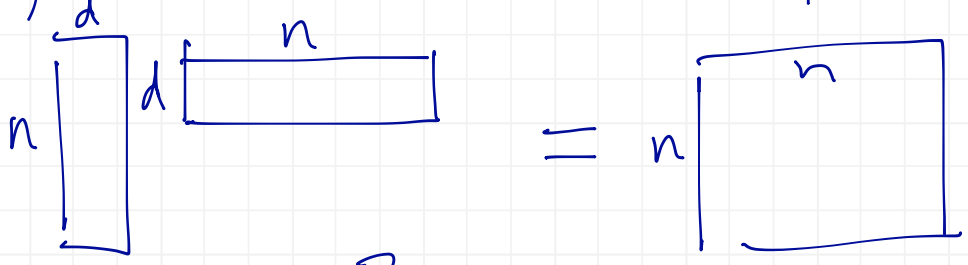
This is a decision tree result, but it is costly to construct the decision tree.

Rest of results on constructing the decision tree

Pf Reduce to n/d Funny MM of $n \times d$ and $d \times n$ matrices



To Funny MM $n \times d$ & $d \times n$ matrix



need $a_{ik} + b_{kj} \overset{?}{\leq} a_{ik'} + b_{k'j}$ $\forall$ pairs k, k'

$\Leftrightarrow$ $a_{ik} - a_{ik'} \overset{?}{\leq} b_{k'j} - b_{kj}$

Just sort all $\underbrace{a_{ik} - a_{ik'}}_{O(d^2 n) \text{ elements}}$ and $\underbrace{b_{k'j} - b_{kj}}_{\text{same}}$

$\Rightarrow O(d^2 n \log(d^2 n))$ comparisons

(After this, need O comparisons, but lots of other work)

$\Rightarrow$ final # comparisons $O\left(\frac{n}{d} d^2 n \log n + \underbrace{\frac{n}{d} n^2}_{\text{final min's}}\right)$

choose $d = \sqrt{n}$ $\square$

(can do a bit better with $d = \frac{\sqrt{n}}{\sqrt{\log n}}$)

Remark implies slightly subcubic algorithm
by building entire decision tree (exponential size) for SMALL input size b
and doing $(n/b)^3$ Funny MM on $b \times b$ matrices

Note recent result on 3SUM uses same idea

Chan '05

digression. A Problem in Computational Geometry: Dominance Searching

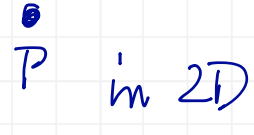
Given n red points P and n blue points Q in $\mathbb{R}^d$

find all pairs (p, q) $p \in P$, $q \in Q$ s.t.

p is <u>dominated</u> by q

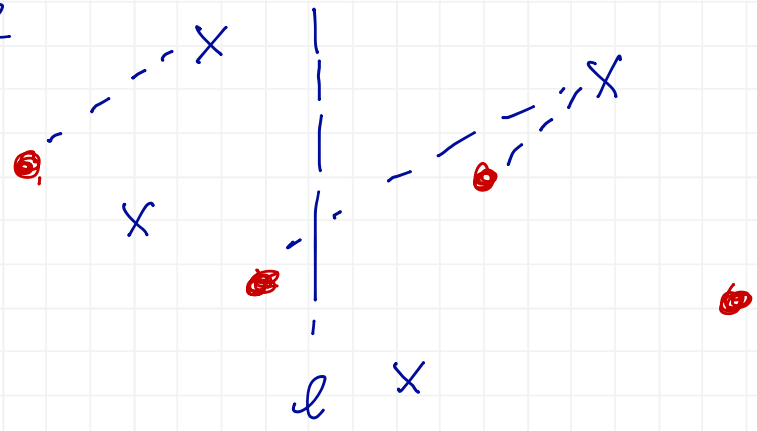
$\Leftrightarrow (p_1 \cdots p_d) \leq (q_1 \cdots q_d)$

i.e. $p_1 \leq q_1 \quad \cdots \quad p_d \leq q_d$



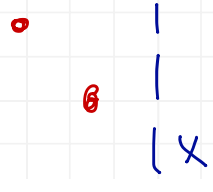
two versions: count / report

Example



Algorithm (exercise in Divide & Conquer)
in $\mathbb{R}^2$

1. Divide by median vertical line l
2. recurse on left
3. recurse on right
4. count dominating pairs between left red points and right blue points



sort by y, scan bottom to top

$\Rightarrow T(n) = 2T(\frac{n}{2}) + \underbrace{O(n \log n)}_{\text{sort to get median \& do step 4}}$

$\Rightarrow O(n \log^2 n)$ for counting.

To shave off one log — sort once up front

For reporting $O(n \log^2 n + X)$ X = output size

in $\mathbb{R}^d$: do line 4 by projection & recursion in $\mathbb{R}^{d-1}$

$\Rightarrow T_d(n) = 2T_d(\frac{n}{2}) + T_{d-1}(n)$

$\Rightarrow O(n \log^d n)$ by induction

(best known: can shave off 3 log factors)

we will use variable d

Compare $\log^d n$ with n

$\log^d n = 2^{d \log\log n}$ $\quad$ $n = 2^{\log n}$

So for $d = .9 \frac{\log n}{\log\log n}$ get $O(n^{1.9})$

"Better" solution to recurrence

(for moderately large d)

Solving recurrence above for $T_d(n)$

Guess $T_d(n) \leq c^d n^{1+\varepsilon}$ ~~for some c, ε~~

induction works when $2c^d \left(\frac{n}{2}\right)^{1+\varepsilon} + c^{d-1} n^{1+\varepsilon} \leq c^d n^{1+\varepsilon}$

$$\Leftrightarrow \frac{c^d}{2^\varepsilon} + c^{d-1} \leq c^d$$

$$\frac{c}{2^\varepsilon} + 1 \leq c$$

$$c = \frac{1}{1 - \frac{1}{2^\varepsilon}}$$ so for any ε, can find c.

e.g. $\varepsilon = \frac{1}{2}$ $c \approx 3.42$

$\Rightarrow O(3.42^d n^{1.5})$

e.g. for $d = .2 \log n$

$\Rightarrow O(3.42^{.2 \log n} n^{1.5}) = O(n^{.2 \log 3.42} n^{1.5})$

(for counting) $= O(n^{1.9})$

Solving APSP (this is new idea of Chan's paper)

To do Funny MM $n \times d$ matrix A
$d \times n$ matrix B

$$c_{ij} = \min_{k=1..d} (a_{ik} + b_{kj})$$

Fix k. For what entries c_{ij} is the min realized by k?

$$\{(i,j): a_{ik} + b_{kj} \leq a_{ik'} + b_{k'j} \quad \forall k' = 1..d\}$$

$$= \{(i,j): a_{ik} - a_{ik'} \leq b_{k'j} - b_{kj} \quad \forall k' = 1..d\}$$

$$= \{(i,j): (a_{ik} - a_{i1}, a_{ik} - a_{i2} \cdots a_{ik} - a_{id})$$
is dominated by $(b_{1j} - b_{kj}, b_{2j} - b_{kj} \cdots b_{dj} - b_{kj})\}$

$\Rightarrow$ dominance searching in $\mathbb{R}^d$
for n red/blue points.

For $d = .2 \log n$ get $O(n^{1.9} + X_k)$ time

(X_k: output size)

This is for one k.

Total time $O(dn^{1.9} + \sum_{k=1}^{d} X_k)$

$= O(n^{1.9} \log n + n^2)$

(because each pair (i,j) appears once)

$= O(n^2)$

Final Time $O(\frac{n}{d} n^2) = O(\frac{n^3}{\log n})$

improvements?

geometric algs tend to have exponential dependence on dimension.

new idea: use different dividing line
not median, but sensitive to
#red, #blue on left/right

math to solve recurrence