Shortest Paths [Lecture 5]

Today: APSP in subcubic time for "large" edge weights, directed graphs.

(1) Fredman's ~~[illegible]~~ decision tree bound

(2) Ryan Williams' $n^3/2^{(\lg n)^{\delta}}$ APSP algorithm.

Can $\exists$ be a "~~truly~~" subcubic algorithm for APSP?

May depend on the model.

- Kerr: if we build a circuit where ~~the~~ gates are add/subtract and min then need $\Omega(n^3)$ size.

- But Fredman [1975] showed how to compute APSP using only $O(n^{2.5} \text{polylog} n)$ comparisons — though the actual time to do the ~~[illegible]~~ APSP was more. Formally: built a decision tree of depth $\tilde{O}(n^{2.5})$, but large size.

- Could use this to get slight improvement: $O(n^3/\sqrt[3]{\log n})$ time. Then sequence of improvements to $O(n^3/\log^2 n)$ or so. [Chan]

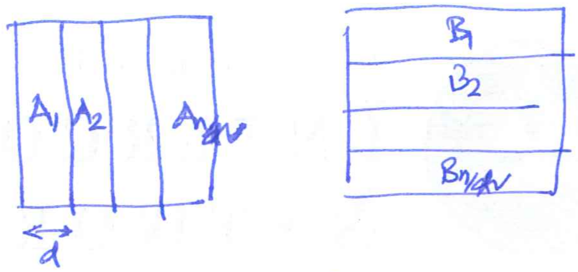
- Recent breakthrough [Williams 2014] using new ideas.

OK: Fredman. Want to compute MinSumProduct $C_{ij} = \min_k \{A_{ik} + B_{kj}\}$

Thm: Can perform $n^{2.5}$ comparisons and then all the C_{ij} values will be determined.

Recall: $APSP(n) = \Theta(MinSumProduct(n))$ [Munro, Fisher-Meyer]

Proof: Idea 1: break A, B into



n/w blocks of width w

$$(A\odot B)_{ij} = \min\left((A_1\odot B_1)_{ij}, (A_2\odot B_2)_{ij} \quad \dots \quad (A_{n/w}\odot B_{n/w})_{ij}\right)$$

each $n\times n$ matrix

$$(A_\ell\odot B_\ell)_{ij} = \min_{\substack{\cancel{k\in[d(\ell-1)+1..d\ell]} \\ k\in[w(\ell-1)+1..w\ell]}} (A_{ik}+B_{kj})$$

Sps each $A_\ell \odot B_\ell$ takes time T

$$\Rightarrow \frac{n}{w}\cdot T + \frac{n}{w}\cdot n^2$$

to compute $A\odot B$ from $(A_\ell \odot B_\ell)$s.

$$\Rightarrow T \ll wn^2 \text{ ideal.}$$

Idea 2: Compute $A'\odot B'$ with $O(w^2 n\log n)$ comparisons.

A': $n\times w$, B': $w\times n$.

Pf: Suppose $(C')_{ij} = A'_{ik^*}+B'_{k^*j}$ for k^*

$$\Leftrightarrow A'_{ik^*}+B'_{k^*j} \le A'_{ik}+B'_{kj} \quad \forall k\in[w]$$

$$\Leftrightarrow A'_{ik^*}-A'_{ik} \le B'_{k^*j}-B'_{kj} \quad \forall k.$$

$\forall k,\ell\in[w]$

take k,ℓ columns of A, k,ℓ rows of B.

and ~~compute~~ sort the $2n$ numbers

$$\{a_{ik}-a_{i\ell}\}_{i\in[n]} \quad \{b_{kj}-b_{\ell j}\}_{j\in[n]}$$

Need perform no more comparisons. [only these $w^2 \times 2n\lg n$ comp]

$$T = w^2 n \lg n.$$

To figure out C_{ij} look for k st. $\forall l$ $a_{ik} - a_{il}$ precedes ~~$b_{jk} - b_{jl}$~~ $b_{kj} - b_{kl}$

determined by these sorted lists

$\Rightarrow$ total comparisons $= T(n/w) + n/w \cdot n^2 = n^2 w \lg n + n^3/w \Rightarrow OPT = n^{2.5}\sqrt{\lg n}.$

Fredman shows how to get rid of $\sqrt{\lg n}$, we'll skip it

---

~~Williams' Algorithm in $\frac{n^3}{2^{(\lg n)^\delta}}$ time for some $\delta \le 1/2$ [Paper shows $\delta = 1/2$].~~

~~How to compute $A \odot B$ in $\tilde{O}(n^2)$ time, for some "large" w? Think of $w \approx 2^{(\lg n)^{1/100}}$.~~

~~{ larger than $(\lg n)^{100}$~~
~~smaller than n^ϵ.~~

~~$A, B \in$ entries say are from $1..M$~~

~~$n \times w$ $\quad$ $w \times n$ $\quad$ $\Rightarrow$ ~~distances are ... n^2 over~~ $\quad$ $\lg_2 M$ bits~~

~~Say just want to compute the bottom bit of $C = A \odot B$ entries. Idea exactly same.~~

~~$C_{ij} = \min_k \{a_{ik} + b_{kj}\}$~~
~~$= a_i \odot b_j$~~

(I) Computing ~~the~~ any bit of $MSP(u, v)$ can be done using a circuit of
- constant depth
- unbounded fan-in
- small size $s \leftarrow (w \log M)^{O(1)} = N^{O(1)}$

$u, v \in [M]^w$

Easy: adds, comps not difficult to do. (HW?)

(II) "Ideal thm" [false]

~~if a~~ can write a polynomial over $\mathbb{F}_2$ $P(\bar{u}, \bar{v})$ ~~$P(u_1, u_2, \ldots u_w, v_1, v_2, \ldots v_w)$~~

$2w \lg M$ vars. in F_2

st $P(\bar{u}, \bar{v}) = \bar{u} \odot \bar{v}$.

first bit of MSP.

P has degree $D \le (\log s)^{O(1)} = (\lg N)^{O(1)}$

$\Rightarrow$ at most $\binom{2N}{D}$ monomials.

So now need to evaluate P on all $(a_i, b_j) \in A \times B$ pairs

[Q: how much time to find this poly?]

$(C_{ij})_{1st\ bit} = P(\bar{a}_i, \bar{b}_j).$

~~Claim: $\exists$ a distribution over polynomials $\{P(\bar{u},\bar{v})\}$ over $2N$ Boolean variables $u_1 \dots u_N$, $v_1 \dots v_N$~~
~~$\in \mathbb{F}_2[u_1 \dots u_N, v_1 \dots v_N]$ $\quad 2N = 2(w \log M)$~~
~~s.t. $\Pr_{P \leftarrow D}[P(\bar{u},\bar{v}) = (u \odot_\ell v)] \geq 3/4$.~~ the ℓ^{th} bit

---

Williams' Algorithm: Same idea $\underset{n\times n}{A} \odot \underset{n\times n}{B}$ broken into n/w pieces of $\underset{n\times w}{A} \odot \underset{w\times n}{B}$.

note that $\underset{\text{scalar}}{C_{ij}} = \underset{1\times w}{a_i} \odot \underset{w\times 1}{b_j} = \min_k \{a_{ik} + b_{kj}\}$.

$\boxed{\text{numbers are integers in } [0..M] \text{ say}}$

let $u \odot_\ell v$ take the min sum product of u & v then take the ℓ^{th} bit.
$= \min_k \{u_k + v_k\}$
if we could compute this for all $\log M$ positions, for all $\binom{u}{v} = (a_i, b_j)$ $\forall ij \in n^2$ we'd be done.

So challenge: compute C^ℓ ~~$\{u \odot_\ell v \mid (u,v) = (a_i, b_j) \in \dots\}$~~
st $C^\ell_{ij} = a_i \odot_\ell b_j$ in time $\tilde{O}(n^2)$ time for w large.
(a_i: $w \log M$ bits, b_j: $w \log M$ bits)

— x —

$\boxed{\text{call } w \log M = N.}$

Idea: (I) find a polynomial $Q(u_1 \dots u_N, v_1 \dots v_N) \in \mathbb{F}_2[x_1 \dots x_{2N}]$
s.t. $Q(\bar{u},\bar{v}) = u \odot_\ell v \quad \forall u,v \in \{0,1\}^N$
interpreted as minsum product of w bit vectors of numbers in $[0..M]$

(II) Evaluate $Q(\cdot,\cdot)$ on all elements in ~~$A \times B$~~ $\{a_i; b_j\}$ ← of size n^2.
in time $\tilde{O}(n^2)$.

— x —

For this we'll need that Q has small degree. $O(D)$.

Let u^S denote $\prod u_i^{\mathbb{1}(i\in S)}$ for $S \subseteq [N]$.
eg $u^{\{1,4,7\}} = u_1 u_4 u_7$.

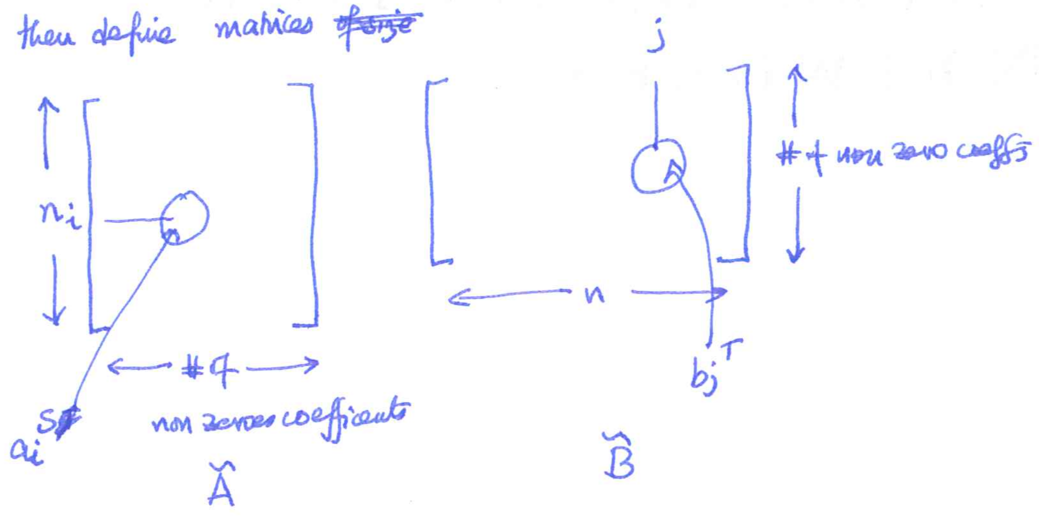
$Q = \sum_{\substack{|S|\leq D \\ |T|\leq D}}' c_{S,T}\, u^S v^T$

because $\mathbb{F}_2$ then $c_{S,T} \in \{0,1\}$

~~similarly v^S denote~~

Idea II first: Sps $Q(u,v) = \sum_{|S|,|T|\le D} c_{S,T}\, u^S v^T$ have low degree.

then define matrices ~~of size~~



$\tilde{C} = \tilde{A} \cdot \hat{B}$ then $\tilde{C}_{ij} = \sum_{\substack{|S|,|T|\le D \\ c_{ST}=1}} a_i^S b_j^T = Q(a_i, b_j) = a_i \odot_\ell b_j$

This operation of n^2 evaluations done by "ordinary" rectangular matrix mult. !!

And! Just like square MM, rectangle MM also has fast algos.

Thm [Coppersmith]: Can multiply $(n \times n^{0.1}) \times (n^{0.1} \times n)$ matrices in $\tilde{O}(n^2)$ time.

So as long as # non zero coefficients $\le n^{0.1}$ we're OK.

$$\text{degree } 2D \quad \binom{2N}{2D} \le (2N)^{2D} \overset{!}{\le} n^{0.1}$$

Recall $N = (w \lg M)$

Fact 2: Want a polynomial representation of $\bar{u} \odot_\ell \bar{v}$.

Thm: $\exists$ a distribution $\mathcal{D}$ over polynomials of degree $2D \le (\log N)^{O(1)}$

such that $\forall u, v \in [M]^w$

$$\Pr_{P \leftarrow \mathcal{D}}\left[P(u,v) = u \odot_\ell v \right] \ge 3/4.$$

Also can sample from this distribution in time $O\left(\binom{N}{2D}\right)$.

$\Rightarrow$ we can sample this polynomial and run the evaluation.

takes $\tilde{O}(n^2)$ time overall. About $3/4$ of entries correct.

Now repeat this $O(\log n)$ times and take the majority. $\Pr[\text{this is incorrect}]$ is very small.

So almost as good as getting a single correct polynomial.

Total runtime : want it to be $\tilde{O}(n^2)$.

but $D = (\log N)^{O(1)}$.

and want $N^{O(D)} \overset{!}{\leq} n^{0.1}$

$$\Rightarrow 2^{(\log N)^{O(1)}} \leq n^{0.1} \Rightarrow \log N \leq (\log n)^{1/c}$$

$$\Downarrow$$

$$N \leq 2^{(\log n)^{\epsilon}}$$

$$\Rightarrow W \leq \frac{1}{\log M} \cdot 2^{(\log n)^{\epsilon}}.$$

$$\Rightarrow \text{APSP time} \leq n^3/W \simeq \tilde{O}\left(n^3 / 2^{(\log n)^{\epsilon}}\right).$$

How to prove polynomial representation theorem?

(A) Show that MSP can be performed by a low depth circuit. (AC_0)

i.e. of size $S = N^{O(1)}$ and constant depth (and unbounded fan-in).

(B) Show that any low-depth circuit (depth $O(1)$, size S) can be probabilistically represented by a polynomial over $\mathbb{F}_2$ of degree $D \leq O(\log S)^{O(1)}$.

Main Idea : OR of $x_1, \dots x_k$. How to do it?