PART III: Advanced Algorithmic Techniques

can we solve APSP in slightly better than n^3 time?
3SUM " " " n^2 ?
OV " " " n^2 ?

Shaving Logs

$$n^d \rightarrow n^d / \log^c n \ ?$$

Ex 1: Boolean Matrix Multiplication

Arlazarov, Dinic, Kronrod, Faradzhev '70

$O(n^3/\log^2 n)$ "Four Russians alg'm"

(worse than Strassen etc., but no algebraic techniques "combinatorial alg'm")

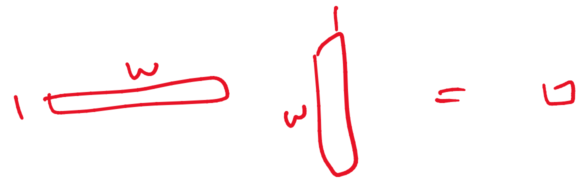
idea 1 - "bit tricks"

standard model of computation
- RAM with w-bit words with $w \geq \log n$
  (s.t. an index/pointer fits in a word)
  standard ops on words in $O(1)$ time

Lemma Can compute Boolean product of an $1 \times w$ and $w \times 1$ matrix in $O(1)$ time. (instead $O(w)$).

an $1 \times w$ ...

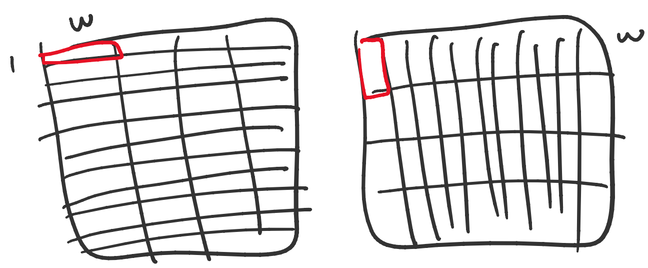
in $O(1)$ time. (instead $O(w)$).



Pf: by bitwise-AND op.

1011   1101   1001 $\square$

To multiply two $n \times n$ matrices:



reduces to $n \cdot \frac{n}{w} \cdot n$ products of $1 \times w$ and $w \times 1$

$$\Rightarrow \quad O\left(n \cdot \frac{n}{w} \cdot n \cdot 1\right) = O\left(\frac{n^3}{w}\right) = \boxed{O\left(\frac{n^3}{\log n}\right)}$$

Rmk - what if bitwise & not supported?

by table lookup
     size of table $O(2^w \cdot 2^w) = O(4^w)$
     time to build $O(4^w \cdot w) \rightarrow O(n^2 \log n)$.

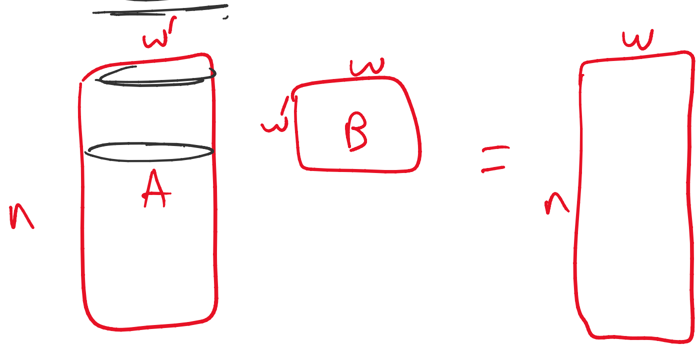
Set $w = \log n$

idea 2 - more table lookup!
     a real "4 Russians" trick

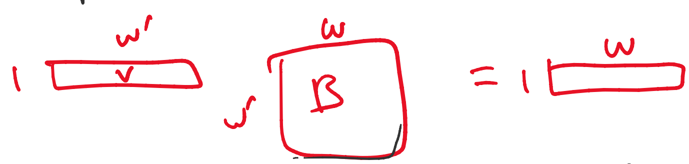
Lemma: can compute Boolean product of ... for $w' = \delta \log n$

Lemma Can compute Boolean product of $n \times w'$ and $w' \times w$ matrices in $O(n)$ time (instead $O(nw'w)$) for $w' = \delta \log n$

Pf:



for each possible row vector v, precompute

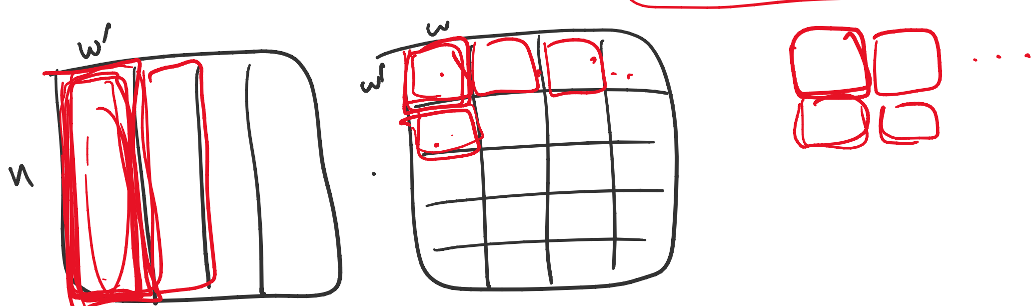


Store in table of size $O(2^{w'})$

building time $O(2^{w'} w' w)$

$= O(n^{\delta} w' w) \ll n.$ ∎

To multiply 2 $n \times n$ matrices, ($\frac{n}{a}\frac{n}{b}\frac{n}{c}$ #'s of $a \times b$ & $b \times c$ products)



$\frac{n}{w'} \cdot \frac{n}{w}$

reduce to $1 \cdot \frac{n}{w'} \cdot \frac{n}{w}$ products of $n \times w'$ and $w' \times w$

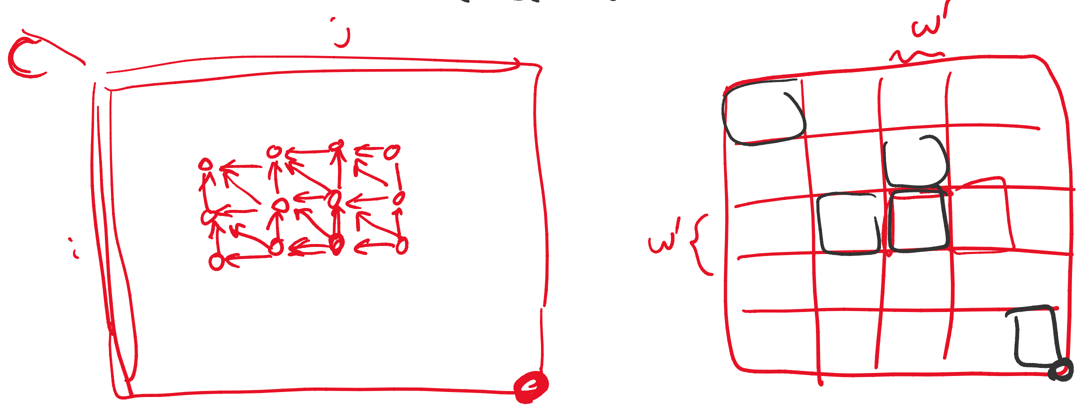
$$\Rightarrow O\left(\frac{n}{w'} \cdot \frac{n}{w} \cdot n\right) = O\left(\frac{n^3}{w' \cdot w}\right) = \boxed{O\left(\frac{n^3}{\log^2 n}\right)}$$

(Bansal–Williams '09 $O(n^3/\log^{2.25} n)$
C. '15 $O(n^3/\log^3 n)$
Yu '16 $O(n^3/\log^4 n)$)

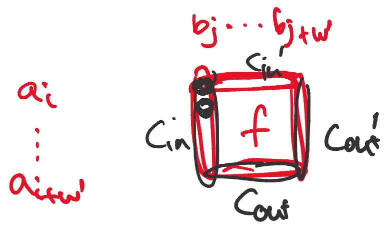
Ex2: LCS / Edit Distance

Masek–Paterson '80 $O(n^2/\log^2 n)$
when alphabet size $\sigma = O(1)$.

$$C[i,j] = \min \begin{cases} C[i-1,j]+1, \\ C[i,j-1]+1, \\ C[i-1,j-1] + d(a_i, b_j) \end{cases}$$



idea – bit tricks / table lookup



define

$f(a, b, C_{in}, C_{in}')$

a, b: strings of length w'

C_{in}, C_{in}': vectors of length w'

$= (C_{out}, C_{out}')$

build table for f:

size of table $= O(\sigma^{w'} \cdot \sigma^{w'} \cdot 2^{w'} \cdot 2^{w'})$

$= O(\sigma)^{w'} \ll n^{O(\delta)}$

$\ll o(n)$.

time to build $= O(\sigma)^{w'} \cdot w'^2 \ll o(n)$

by setting $w' = \delta \log_\sigma n$

$\Rightarrow$ time $O\left(\left(\frac{n}{w'}\right)^2\right) = \boxed{O\left(\left(\frac{n}{\log_\sigma n}\right)^2\right)}$

Ex3: 3SUM for ints

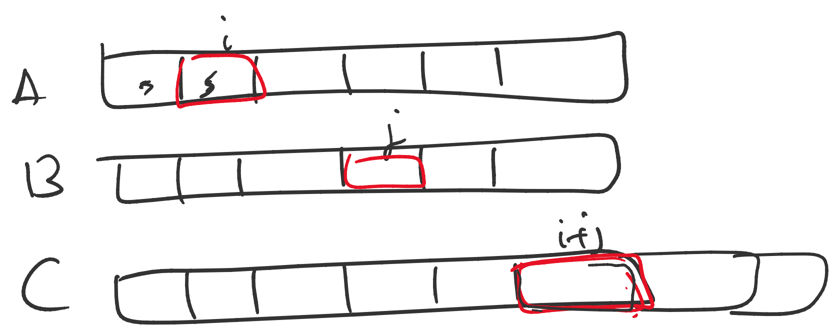
Baran, Demaine, Patrascu '05 $O\left(\frac{n^2 (\log\log n)^2}{\log^2 n}\right)$

idea - hashing again!

$h(x) = x \bmod p$ with rand. prime $p \in (\frac{R}{2}, R)$ for a small R.

Warm-up with Convol3SUM:

divide into blocks



$\Rightarrow$ $O\left(\left(\frac{n}{s}\right)^2\right)$ subprobs of $O(s)$ numbers

after hashing, $O(s \log R)$ bits

by table lookup $\rightarrow$ $O(s \log R)$ bits

size of table
time to build $\rightarrow$ $2^{O(s\log R)} = 2^{O(s\log s + s\log\log n)} \ll n$

for $a_i + b_j \neq c_{i+j}$, $\Pr\left[h(a_i) + h(b_j) = h(c_{i+j}) \text{ or } h(c_{i+j}) - p\right]$

$$\leq \tilde{O}\left(\frac{1}{R}\right)$$

$\Rightarrow$ expected # false positives $= \tilde{O}\left(\frac{n^2}{R}\right)$

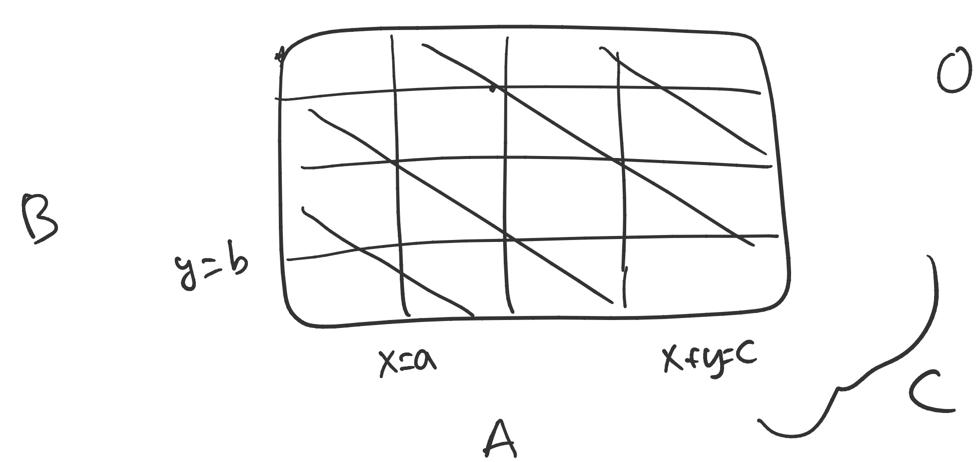
$\Rightarrow$ time to check theses is
$$\tilde{O}\left(\frac{n^2}{R} \cdot s^2\right) \ll O\left(\frac{n^2}{\log^2 n}\right)$$

Set $R = s^2 \log^c n$, $\quad s = \delta \frac{\log n}{\log\log n}$

$$\Rightarrow \quad O\left(\left(\frac{n}{s}\right)^2\right) = \boxed{O\left(\left(\frac{n\log\log n}{\log n}\right)^2\right)}$$

for 3SUM:

Sort A, B, C
then divide into blocks of size s



$O\left(\left(\frac{n}{s}\right)^2\right)$ cells

$O\left(\left(\frac{n}{s}\right)^2\right)$ subprobs of size s

rest is similar.

What about 3SUM for reals?
APSP for reals?