November 13, 2020 10:55 AM

Shaving Logs

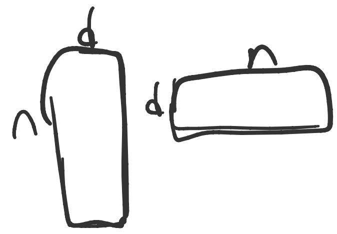
APSP in $O((n^3/\log^{1/5} n)\log^5\log n)$ ← Fredman '75

3SUM in $O((n^2/\log^{1/3} n)\log^5\log n)$ ← Grønlund-Pettie '14

for real #'s

Slightly Faster APSP

$O(n^3/\log n)$ [C '05]



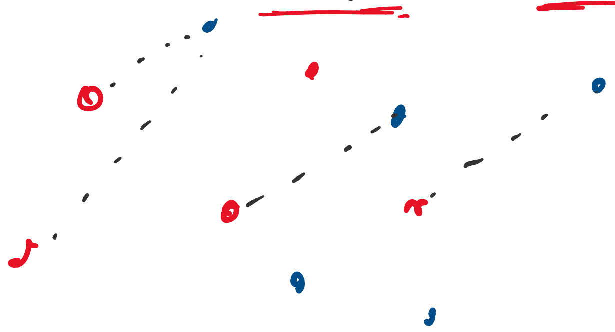
idea - by geometry in $\log n$ dims

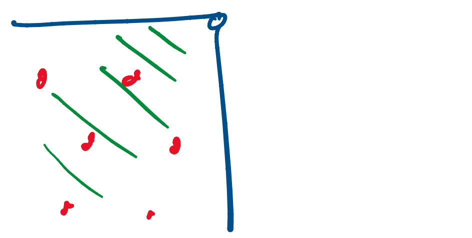
Lemma (from CG)

Given n red pts & blue pts in $\mathbb{R}^d$, can report all K dominating pairs (p,q), p red, q blue s.t. $p_1 \le q_1, p_2 \le q_2, \dots, p_d \le q_d$

in $O(n\log^d n + K)$ time.

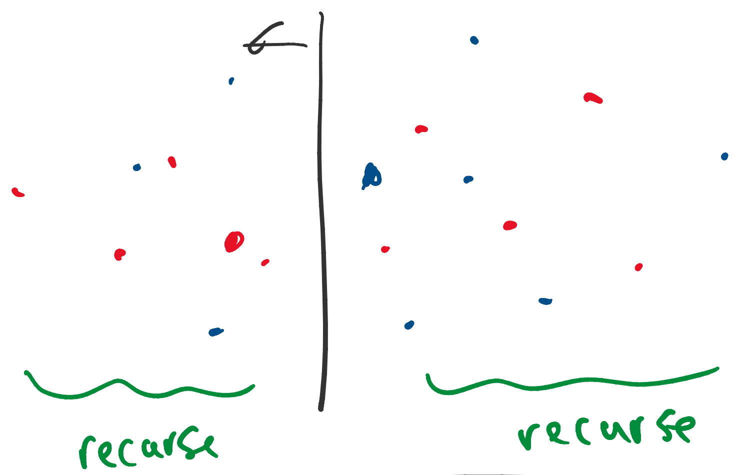
$[C\ O(n\log^{d-3} n + K)]$

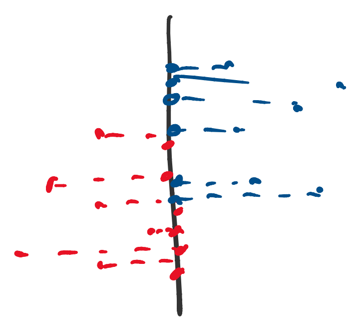




Pf: By D&C.

Take median 1^{st} coord.





$$\boxed{T_d(n) \le 2\,\underbrace{T_d\left(\tfrac{n}{2}\right)}_{\text{red left, blue left or red right, blue right}} + \underbrace{T_{d-1}(n)}_{\text{red left, blue right}} + O(n)}$$

$$T_1(n) = O(n \cancel{\log n})$$

$$T_2(n) = 2T_2\left(\tfrac{n}{2}\right) + O(n\log n) \Rightarrow O(n\log^2 n)$$

$$T_3(n) = 2T_3\left(\tfrac{n}{2}\right) + O(n\log^2 n) \Rightarrow O(n\log^3 n)$$

$$\vdots$$

$$T_d(n) = O(n\log^d n) + K \qquad \square$$

(useful when $d \ll \frac{\delta \log n}{\log\log n}$)

($\log^d n = 2^{d\log\log n}$)

Slightly better analysis:

$$t_d(n) = \frac{T_d(n)}{n} \qquad \text{(cost per pt).}$$

$$\Rightarrow t_d(n) \le \underline{t_d\left(\tfrac{n}{2}\right)} + \underline{t_{d-1}(n)} + \cancel{O(1)}$$

$\Rightarrow$ Counting binary strings of length $\log n + d$ with $\log n$ 0's, d 1's

$$\Rightarrow t_d(n) \le \binom{\log n + d}{d}$$

$$\Rightarrow T_d(n) = O\left(n\binom{\log n + d}{d} + K\right).$$

$$\binom{m}{k} \le \left(\frac{em}{k}\right)^k \qquad = O\left(n\cdot O\left(\frac{\log n}{d}+1\right)^d + K\right)$$

when $d \le \delta \log n$ $\qquad = O\left(\underbrace{n\, O\left(\tfrac{1}{\delta}\right)^{\delta\log n}}_{n^{1+\delta'}} + K\right).$ ✓

Lemma can compute (min,+)-MM of matrix A, B

Lemma can compute (min,+)-MM of $n\times d$ & $d\times n$ matrix A, B in $O(n^2)$ time if $d=\delta\log n$. (instead of $O(dn^2)$).

Pf: Fix $k_0\in[d]$.

Want to compute $\min_{k\in[d]}(a_{ik}+b_{kj})$ for every i,j.

Subproblem determine all i,j s.t.

$$\operatorname{argmin}_{k\in[d]}(a_{ik}+b_{kj}) = k_0.$$

i.e. $a_{ik_0}+b_{k_0j} \le a_{ik}+b_{kj}$ for all $k\in[d]$

Fredman's trick!

i.e. $a_{ik_0}-a_{ik} \le b_{kj}-b_{k_0j}$ $\forall k\in[d]$

i.e. red point $(a_{ik_0}-a_{i1}, a_{ik_0}-a_{i2}, \dots, a_{ik_0}-a_{id})$ dominated by blue point $(b_{1j}-b_{k_0j}, b_{2j}-b_{k_0j}, \dots, b_{dj}-b_{k_0j})$

$\Rightarrow$ report dominating pairs between n red pts & n blue pts in $\mathbb{R}^d$!!

by Lemma, $O(n^{1+\delta'}+K_{k_0})$ for $d=\delta\log n$

Repeat for all $k_0\in[d]$

total time $O(dn^{1+\delta'}+n^2)$

$$= O(n^2). \qquad \square$$

To (min,+)-multiply 2 $n\times n$ matrices:

reduce to $\frac{n}{d}$ products of $n\times d$ & $d\times n$

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

$\Rightarrow O\left(\frac{n}{d} \cdot n \quad\right) = \boxed{O\left(\frac{\quad}{\log n}\right)}.$

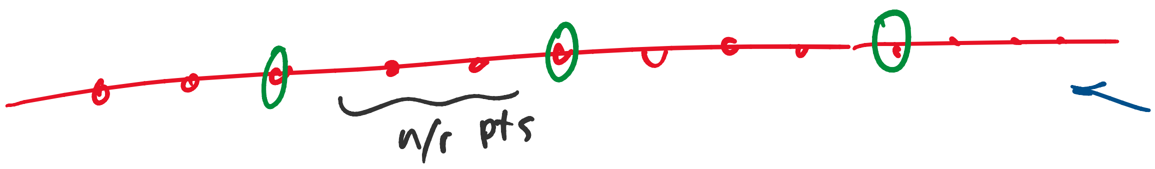
Rmk: combine geometry with bit packing $\Rightarrow \sim n^3/\log^2 n$ [C.'07]

Lopsided D & C $\Rightarrow \sim \underline{n^3/\log^3 n}$ [C.'17]
$(d \sim \log^2 n)$

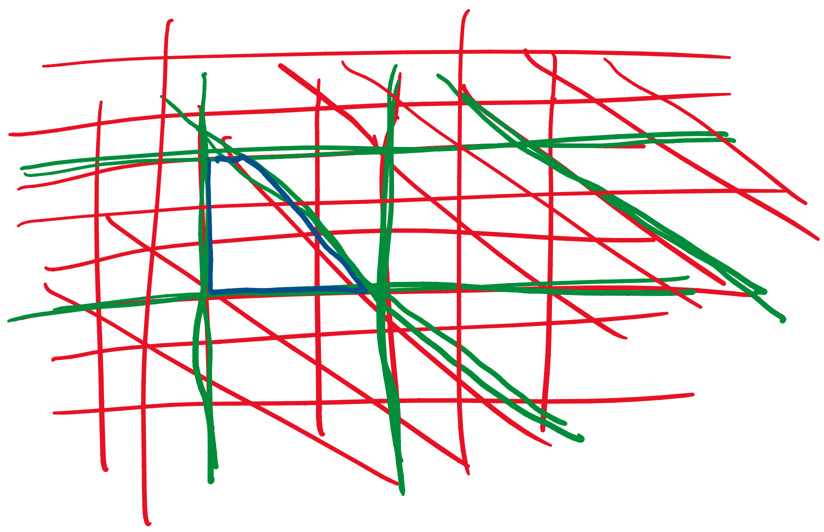
Slightly Faster 3SUM (for Reals)

Lemma (from CG) Given n hyperplanes in $\mathbb{R}^d$,
can cut $\mathbb{R}^d$ into $O(d^{O(d)} r^d)$ cells s.t.
each cell intersects $O(\frac{n}{r})$ hyperplanes.

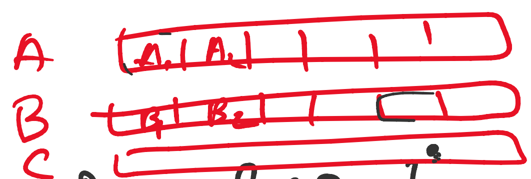
1D



2D



To solve 3SUM for A, B, C:



Sort A, B, C
divide A into blocks $A_1, \ldots, A_{n/d}$ of size d
B . . . $B_1, \ldots, B_{n/d}$

map each block A_i to
point $(A_i[1], \ldots, A_i[d])$ in $\mathbb{R}^d$
[illegible] hyperplanes

point $(A_i[1], \dots, A_i[d])$

map each block B_j to $O(d^4)$ hyperplanes

$$\{(x[1], \dots, x[d]) : x[s] + B_j[t] = x[s'] + B_j[t']\}$$

for each $s, s', t, t' \in [d]$.

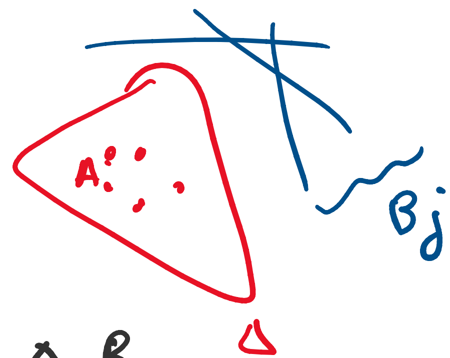
Obs If we know which side A_i is on for all these hyperplanes, know sorted order of $A_i + B_j = \{A_i[s] + B_j[t] : s, t \in [d]\}$

Apply Lemma to these $O(d^4 \cdot \frac{n}{d}) = O(d^3 n)$ hyperplanes.

For each cell Δ, each B_j.

Case 1. if hyperplanes of B_j do not intersect Δ:

all A_i in Δ have the same sorted order of $A_i + B_j$

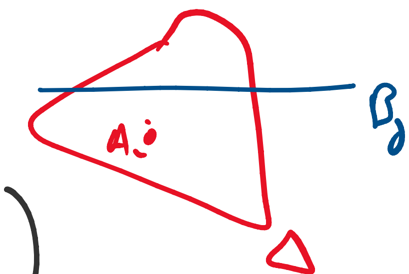


Suffice to sort once per Δ, B_j

$$\Rightarrow O\left(d^{O(1)} r^d \cdot \frac{n}{d} \cdot d^2 \log d\right)$$

Case 2. if a hyperplane of B_j intersects Δ:

Sort $A_i + B_j$ for each A_i in Δ



$$\Rightarrow O\left(\underbrace{n}_{\#A_i\text{'s}} \cdot \underbrace{\frac{d^3 n}{r}}_{\#B_j\text{'s}} \cdot d^2 \log d\right)$$

For each c_k, binary-search in $A_i + B_j$ for $O(\frac{n}{d})$ (i,j) pairs

$$\Rightarrow O\left(n \cdot \frac{n}{d} \cdot \log d\right)$$

Total time: $O\left(d^{O(d)}\, r^d\, \frac{n}{d} \cdot d^2 \log d + n \cdot \frac{d^3 n}{r} \cdot d^2 \log d + n \cdot \frac{n}{d} \log d\right)$

Set $r = d^{10}$: $O\left(\underline{d^{O(d)}}\, n + \underline{\frac{n^2}{d} \log d}\right)$

$2^{O(d \log d)} n$

Set $d = \frac{\delta \log n}{\log \log n}$

$$\boxed{O\left(\frac{n^2}{\log n} (\log\log n)^2\right)}$$

[C.'18]

Rmk: - combine geometry with bit packing

$\Rightarrow O\left(\frac{n^2}{\log^2 n} (\log\log n)^{O(1)}\right)$.

current best.