August 27, 2024 11:59 AM

CS 498 TC: Computational Geometry

http://courses.grainger.illinois.edu/cs498tcu

Course Work:

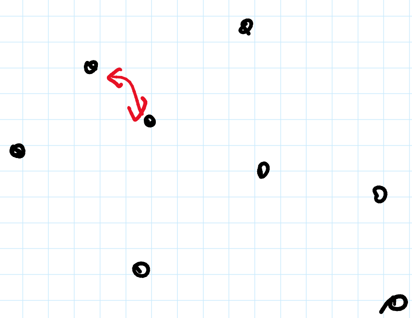
		TCU/TC3	TC4
5 HWs	(groups ≤ 3)	40%	35%
Midterm	(Oct 7 Mon 7p-9p)	20%	15%
Final		40%	35%
(Presentation for TC4 grad.)			15%

Prerequisite: CS374 or equiv.

What is CG?

alg'ms for geom problems

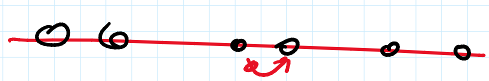
e.g. closest pair



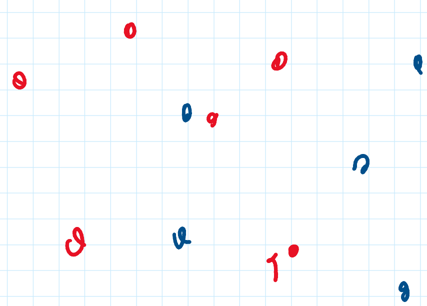
1D: $O(n \log n)$ time

2D: $O(n \log n)$ time

3D & beyond: $O(n \log n)$



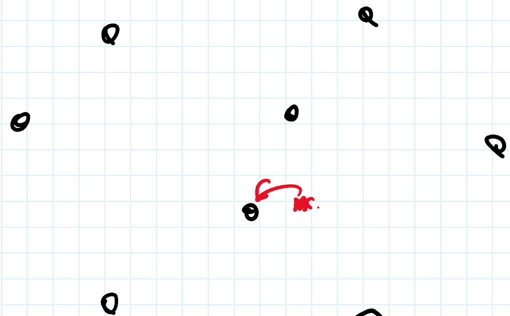
red-blue closest pair



2D: $O(n \log n)$

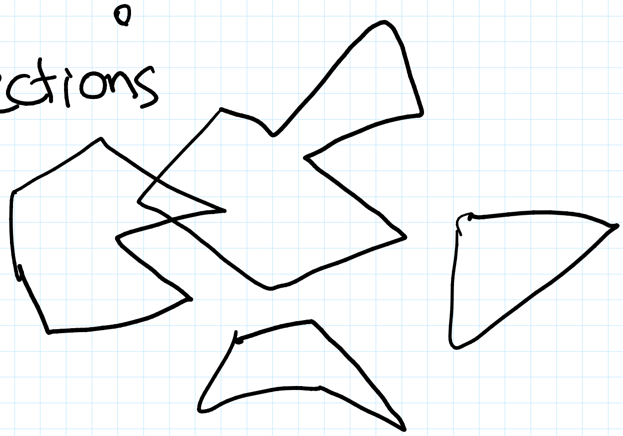
3D: $O(n^{4/3})$

nearest neighbor search



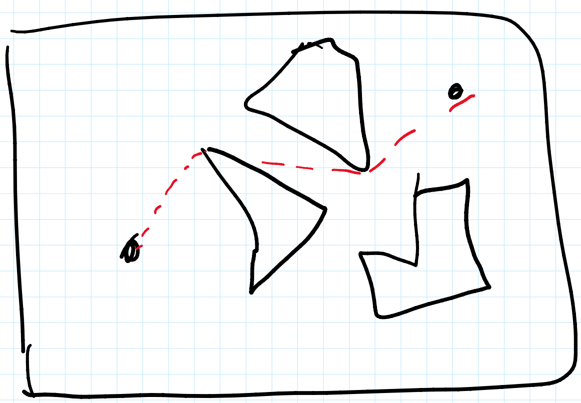
2D: $O(n)$ space
$O(\log n)$ query time

intersections

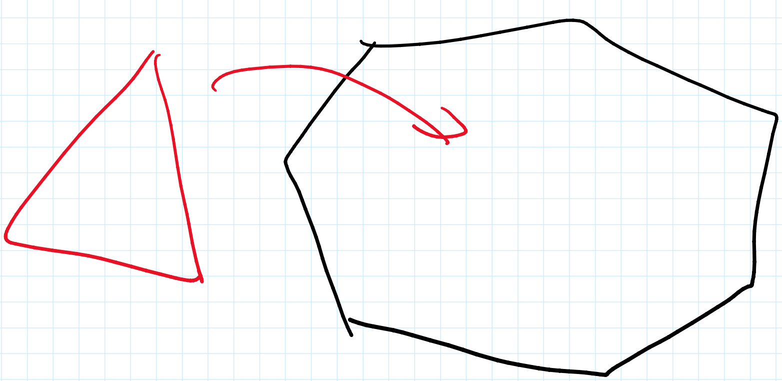


$O(n \log n)$

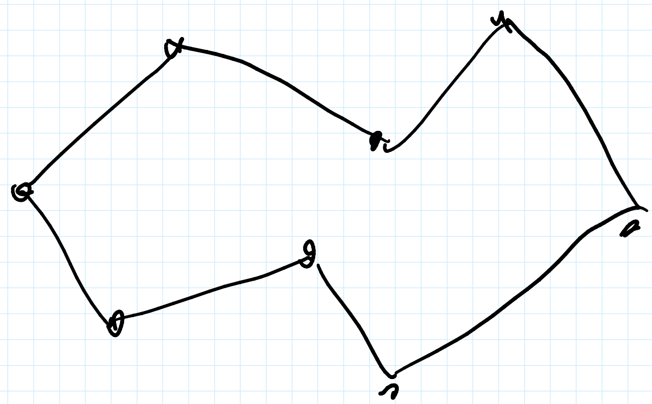
paths



polygon containment



TSP

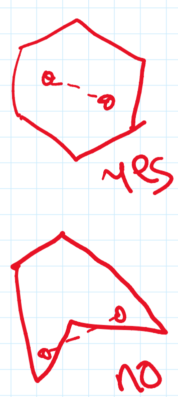


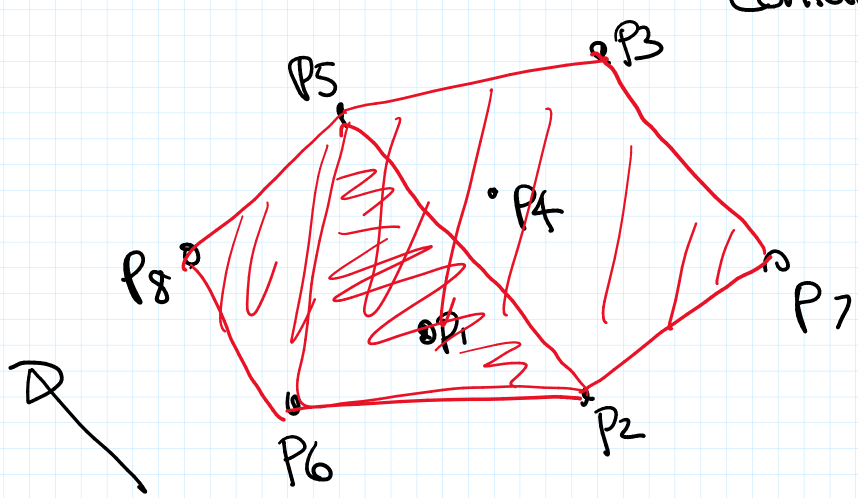
Convex Hull in 2D

$p_i = (x_i, y_i)$

Problem Given n points $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$, compute $CH(P)$ = smallest convex set containing P

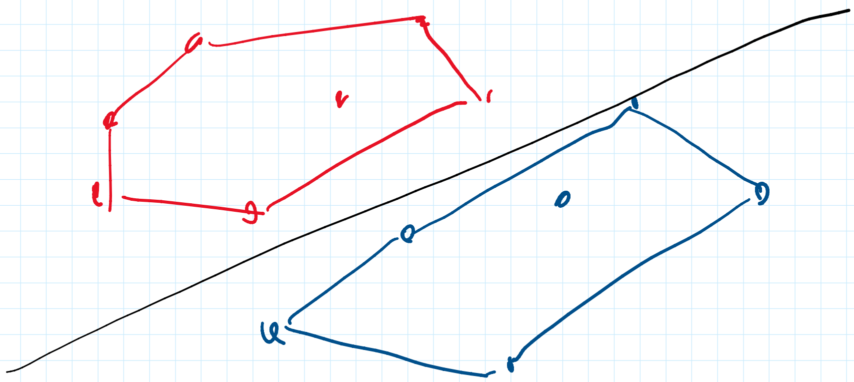
convex polygon (all angles $\leq 180°$)





Output: $p_8\ p_6\ p_2\ p_7\ p_3\ p_5$ (in ccw order) appl - find "shape" of point cloud

- bounding volume
- extreme pts, farthest pair
- line separation



Math Properties

$$CH(P) = \bigcap_{\text{convex } S \supseteq P} S = \bigcap_{\text{halfplane } h \supseteq P} h$$

$$= \{ \text{all convex combinations of } p_1, \ldots, p_n \}$$

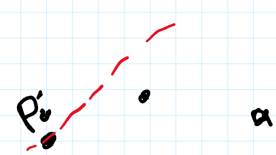
$\sum_{i=1}^{n} \alpha_i p_i$ for some $\alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i = 1$.

$$= \bigcup_{i,j,k} \{ \text{all convex combs of } p_i, p_j, p_k \}$$

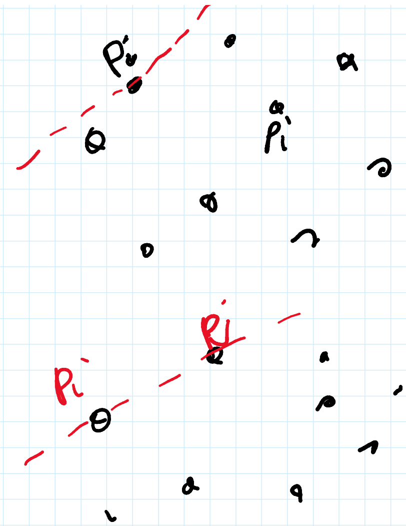
p_i is a vertex of $CH(P)$

iff $\exists$ line thru p_i s.t.

P lies on one side

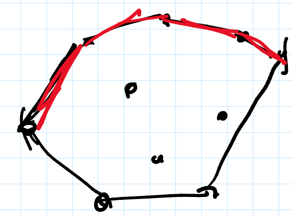


iff $\exists$ line thru p_i s.t.
P lies on one side



$p_i p_j$ is an edge of $CH(P)$
iff P lies on ~~sne~~ one side
of line $\overleftrightarrow{p_i p_j}$

Note: suffice to compute
upper hull (UH)



Alg'm ① : Brute force

for $i = 1$ to n do
    for $j = 1$ to n do {
        flag = true
        for $k = 1$ to n do $(k \neq i, j)$
            if p_k above $\overleftrightarrow{p_i p_j}$ then flag = false
        if flag then output $p_i p_j$
    }

$O(n^3)$ time

Implementation issues:

(i) Primitive ops:
how to test p_k above $\overleftrightarrow{p_i p_j}$?

(ii) degeneracies

(iii) precision issues

⋮