Computational Geometry

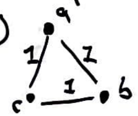
Brittany.fasy@montana.edu

(406-994).4804

What is CG?

- algorithms w/ geometric input/output
  - → I/O is naturally discrete
  - NOT: taking a cont. thing & discretizing to estimate a solution.
  - IS: problems you can talk about & make progress on in A bar on a napkin.
  - e.g., how can I arrange n points on a plane ($\mathbb{R}^2$) to maximize the # of pairs unit distance apart?

(2)


(3)


(4) Forbidden config.


5 pairs < 6 total pairs


cur best upper bound $O(n^{4/3})$

Open: can we do tighter UB?

Words assoc. w/ CG

- Optimization / lin. prog.
- data → clustering
  → machine learning
  → search
- advanced algorithms
- applications: graphics

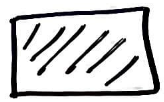
- In CG, we evaluate our solutions
  - → asymptotic worst case RT (sometimes average-case RT)
  - → value efficiency
    $\Theta(n)$ preferred over $\Theta(n \log n)$, which is pref. over $\Theta(n^2)$
    $\Theta(n^3)$ ...
  - → correctness: the algo. must do the task at hand.
    if approximating, we give bounds on how off our sol'n is.

e.g: Quicksort
worst-case: $\Theta(n^2)$
average: $\Theta(n \log n)$

- Dimension: we will focus on 2D / 3D
  historically, much of CG is low-d (≤ 10)
  curse of dimensionality: complexity often grows exponentially w/ dimension

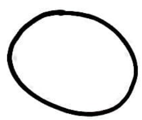
More about CG

- I/O is discrete + geometric,
  often (not always) flat (= defined by affine sets)

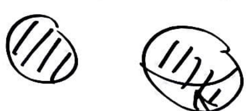
  → $\mathbb{R}^2$, rectangle in $\mathbb{R}^2$ 

  edges = line segments 

  points, lines, (hyper) planes

our non-flat objects { sphere / circle   $\subseteq \mathbb{R}^3$

balls / discs



SOLID

$S^2 := \{x \in \mathbb{R}^3 \mid \|x\|_2 = 1\}$

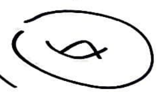
$\mathbb{R}^3 = \mathbb{R} \times \mathbb{R} \times \mathbb{R}$

$x = (x_1, x_2, x_3)$

$S^2 := \{x \in \mathbb{R}^3 \mid \sqrt{x_1^2 + x_2^2 + x_3^2} = 1\}$

"the set of things unit dist from origin

Not focused on { algebraic surfaces



- analysis

— RAM =
a given
— memory
• to

— thing
add
mul
div
sub
$\sqrt{\ }$

* avoid
→(

affine sets)

$\|x\|_2 = 1\}$

$\mathbb{R}$

$, x_3)$

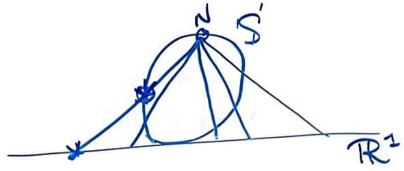
$3 \mid \sqrt{x_1^2 + x_2^2 + x_3^2} = 1\}$

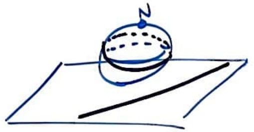
things unit dist from origin

- analysis uses real-RAM MOC

real: working w/ real #s

RAM: random access machine





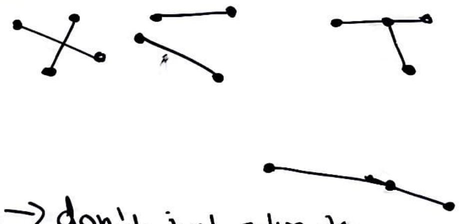
– RAM ⇒ $\Theta(1)$ time to access (R/W) a given location in memory

– memory stores reall #s

- to store a number in $\mathbb{R}^2$: need 2 locations in memory
  - ✓ still $\Theta(1)$

" in $\mathbb{R}^n$: need n locations

Do 2 line segments intersect?



→ don't just solve the equation by setting 2 lines =

– things I can do in const. time:

add 2 #'s

multiply

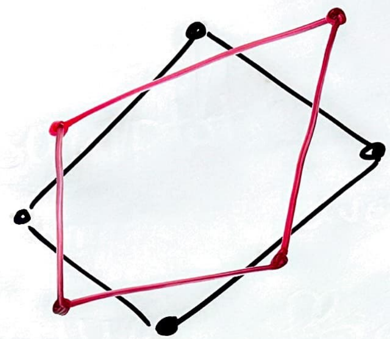
divide

subtract

$\sqrt{\ }$

* avoids dealing w/ floating point arithmetic

→ CGAL simulates real-RAM on many ops!



Q: are the lengths of these 2 the same or different?

Top

1. Co

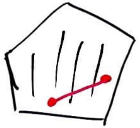
Con

-C

2. Im

TOPICS

1. Convex Hulls



CONVEX

NOT

↳ def'n: line seg. btween any 2 pts
stays inside the shape.

- CH: given a set of pts P,
the CH is the smallest convex
shape containing P.

2. Intersections

3. Triangulations + Subdivisions

4. Point Location

5. Linear Programming

6. Duality pt - line (in $\mathbb{R}^d$)

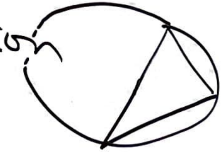
(a,b) $ax+by=1$

Sometimes think of
this as $(a,b,1)$

7 Voronoi Diagrams (post office query
data structure)

↕ duals

8. Delaunay Triangulation



9. Line/Plane arrangements