

Computational Geometry

Intro

15-750
2/1/10
v2

We have been in 1D!

eg sorting, searching, priority queues

These lectures move to 2D.

Apps (even for 2D)

Graphics

Robotics

Geo info systems

CAD/CAM

Sci Comp

Basic approach

Build complex obj out of simple objs.

eg Image $\equiv$ array of dots

Integrated Circuit $\equiv$ planar tri

Basic alg design apps:

1) Divide-and-conquer

eg merge sort

eg quick sort

2) Sweep-Line

3) Random-Incremental

Topic

1) Intro primitive/atomic ops.

2) Sweep-line for line seg intersection

3) 2D Convex hull

4) 2D linear programming

Abs obj	Repre
Real Number	floatin point, big number
Point	Pair of Reals
Line	Pair of points
Line Segment	Pair of points
Triangle	Triple of points

Using points to generate obj's

3

Suppose $P_1, \dots, P_k \in \mathbb{R}^d$

Linear Comb

$$\text{Subspace} = \sum \alpha_i P_i \text{ for } \alpha_i \in \mathbb{R}$$

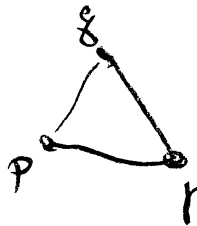
Affine Comb

$$\text{Plane} = \sum \alpha_i P_i \text{ st } \sum \alpha_i = 1 \text{ \& } \alpha_i \in \mathbb{R}$$

Convex Comb

$$\text{Body} = \sum \alpha_i P_i \text{ st } \sum \alpha_i = 1 \text{ \& } \alpha_i \geq 0$$

$\xrightarrow{\text{eg}}$


$$= \{ \alpha p + \beta q + \gamma r \mid \alpha + \beta + \gamma = 1 \text{ \& } \alpha, \beta, \gamma \geq 0 \}$$

Primitive Ops.

4

1) Equality $P = Q$?

2) Line seg intersection test

3) Line side test

Input: (P_1, P_2, P_3)

Output: True if P_3 is "left" of ray $P_1 \rightarrow P_2$

4) In circle test

Input: (P_1, P_2, P_3, P_4)

Output: True if P_4 in circle (P_1, P_2, P_3)

2) Segs $L_1 = [P_1, P_2]$ & $L_2 = [P_3, P_4]$

Let $P_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$

$$\begin{pmatrix} 1 & 1 \\ P_1 & P_2 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ P_3 & P_4 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_3 \\ \alpha_4 \end{pmatrix}$$

st $\alpha_1 + \alpha_2 = 1$

$$\alpha_3 + \alpha_4 = 1$$

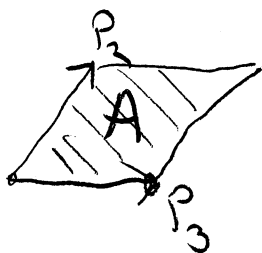
$$\alpha_1, \dots, \alpha_4 \geq 0$$

solve
$$\begin{pmatrix} x_1 & x_2 & -x_3 & -x_4 \\ y_1 & y_2 & -y_3 & -y_4 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

check if $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \geq 0$

3) Line side test

Assume that $P_1 = 0$



$$\det = \begin{pmatrix} x_2 & x_3 \\ y_2 & y_3 \end{pmatrix} \equiv \pm \text{Area of } A.$$

eg $\det \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = -1$ ie P_3 is right of $P_1 \rightarrow P_2$

Claim $LST(P_1, P_2, P_3) = \det \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{pmatrix}$

Pf $RHS = \det \begin{pmatrix} x_1 & x_2' & x_3' \\ y_1 & y_2' & y_3' \\ 1 & 0 & 0 \end{pmatrix} = \det \begin{pmatrix} x_2' & x_3' \\ y_2' & y_3' \end{pmatrix}$

$$P_2' = P_2 - P_1, \quad P_3' = P_3 - P_1$$

Line Segment Inter Prob

Input: n -line seg

Output: All I intersections

Naive: $O(n^2)$ (This is worst case optimal)

Goal: Output sensitive alg

$O(n \log n + |I|)$ today $O((n + |I|) \log n)$

Worst: $|I| = \Omega(n^2)$

Application: Map overlay

Alg today: sweepline

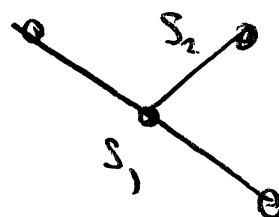
Optimal: Incremental

Sweep Line Alg

$S = \{S_1, \dots, S_n\}$ Segments

Assume: 1) no horizontal segs

2) cases not handled



≤ 3 seg
at a point

Let P = Seg endpoints

I = Seg intersections

Events = $P \cup I$

l = hori line disjoint from $P \cup I$

linear order $\{s \in S \mid s \cap l \neq \emptyset\}$

Order only changes at $P \cup I$

Store the order in Balanced BST.

Idea: Sweep l top to bottom
stopping at events. (Just after)

Problem: We do not know I !

Solution: Compute I just-in-time.

Claim If next event is $s \cap s'$ then s & s' are
neigh.

Keep a priority queue Q_l of events.

Inductively: Q_l contains

1) P below l .

2) Neig inter below l .

Alg Insert P into Q

While $Q \neq \emptyset$

$P = \text{Extract Max}(Q)$

Handle Event(P)

Handle Event(P)

1) if P is upper end of S then

insert(S, D)

add-inter(left(S), S, Q)

add-inter(S, right(S), Q)

2) if P is lower end of S then

add-inter(left(S), right(S), Q)

delete(S, D)

3) if $P \in S \cap S'$

swap(S, S')

add-inter(left(S'), S', Q)

add-inter(S, right(S), Q)

report P

cost	#
$\log n$	n
$\log n$	n
$\log n$	n
$\log n$	n
$\log n$	n
$\log n$	I
$\log n$	I
$\log n$	I

$O((n+|I|)\log n)$