Computational Geometry Intro

15-750
2/22/19

So far we have worked in 1D.
eg sorting, searching, and priority queues
The next few lectures consider 2D.

Apps (even for 2D)	more dim
Graphics	3D Sci Comp
Robotics	Machine Learning
Geo Info Systems	
CAD/CAM	
Sci Comp	

Basic Approach

Build complex objects out of simple objects.

eg Image ≡ array of dots
Integrated Circuit ≡ collection of triangles

Basic Alg Design Approaches

1) Divide-and-conquer

   eg Merge sort

   eg Quick Sort

2) Sweep-Line
3) Random-Incremental

Topics to cover

1) Intro Primative/atomic ops
2) Sweep-line for line seg intersection
3) 2D Linear Programming
4) 2D Convex Hull
5) Many More

Abstract Objs	Repre
Real Number	floating point, big number
Point	Pair of Reals
Line	Pair of Points
Line Segment	Pair of Points
Triangle	Triple of Points

Using Points to Generate Objects

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

Linear Combinations

Subspaces $\equiv \sum \alpha_i P_i$ for $\alpha_i \in \mathbb{R}$

Affine Combinations

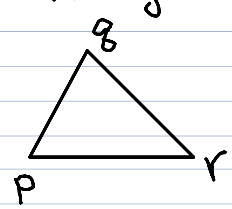
Plane, HyperPlanes $= \sum \alpha_i P_i$ st

$\alpha_i \in \mathbb{R}$ & $\sum \alpha_i = 1$

Convex Combinations

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

eg Triangle



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

Primative Ops.

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 $P_1 \to P_2$

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 $S_1 = [P_1, P_2]$ & $S_2 = [P_3, P_4]$

Let $P_i = \begin{pmatrix} X_i \\ Y_i \end{pmatrix}$

$$\begin{pmatrix} | & | \\ P_1 & P_2 \\ | & | \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = \begin{pmatrix} | & | \\ P_3 & P_4 \\ | & | \end{pmatrix} \begin{pmatrix} \alpha_3 \\ \alpha_4 \end{pmatrix} \quad \text{iff } L_1 \cap L_2 \neq \emptyset$$

st
$$\alpha_1 + \alpha_2 = 1$$
$$\alpha_3 + \alpha_4 = 1$$
$$\alpha_1, \ldots, \alpha_4 \geq 0$$

<u>Case 1</u> $L_1 = \text{Line}[P_1, P_2]$ & $L_2 = \text{Line}[P_3, P_4]$ intersect at unique point.

Let $A = \begin{pmatrix} x_1 & x_2 & -x_3 & -x_4 \\ y_1 & y_2 & -y_3 & -y_4 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$ then A is non singular

Solve $A \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$ $\quad (*)$

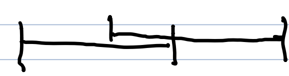
Return: true iff $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \geq 0$

<u>Case 2</u> $L_1 \cap L_2 = \emptyset$

In this case A is singular & no solution to $(*)$.

Return: false.

<u>Case 3</u> $L_1 \cap L_2 = \text{Line}$



1) A is singular

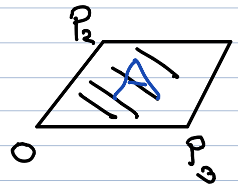
2) $L_1 = L_2$

3) Solve a 1D line seg intersection prob.

---

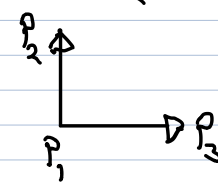
3) Line side test:

Assume that $P_1 = 0$



Claim: $\det\begin{pmatrix} x_2 & x_3 \\ y_2 & y_3 \end{pmatrix} = \pm$ Area of A

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



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

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

We just tranlated P_1 to origin

Line Segment Inter Prob

Input: n-line segs
Output: All I intersections

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

Goal: Output sensitive alg

Known $O(n \log n + |I|)$ **Today** $O((n+|I|) \log n)$

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

Application: Map overlay

Alg today: sweepline

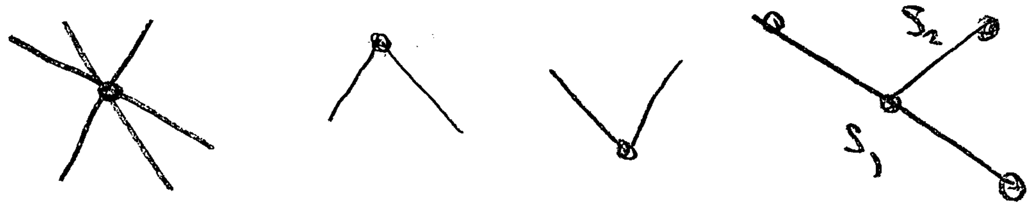
Optimal: Random Incremental

Sweep Line Alg

$S = \{s_1, \dots, s_n\}$ Segments

Assume: 1) no horizontal segs

2) cases not handled today!



≥ 3 seg at a point

Let $\bar{P} \equiv$ Seg end points

$I \equiv$ Seg intersections

Events $\equiv \bar{P} \cup I$

ℓ = hori line disjoint from $\bar{P} \cup I$

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

<u>Note</u> Order only changes at $\bar{P} \cup I$

Store the ordered segs in Balanced BST, D.

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

Problem: We do not know I!
Solution: Compute I just-in-time.

<u>Claim</u> 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 SweepLine($S_1, \dots, S_n$)

Insert $\vec{P}$ into Q

While $Q \neq \emptyset$

a) P = ExtractMax(Q)

b) HandleEvent(P)

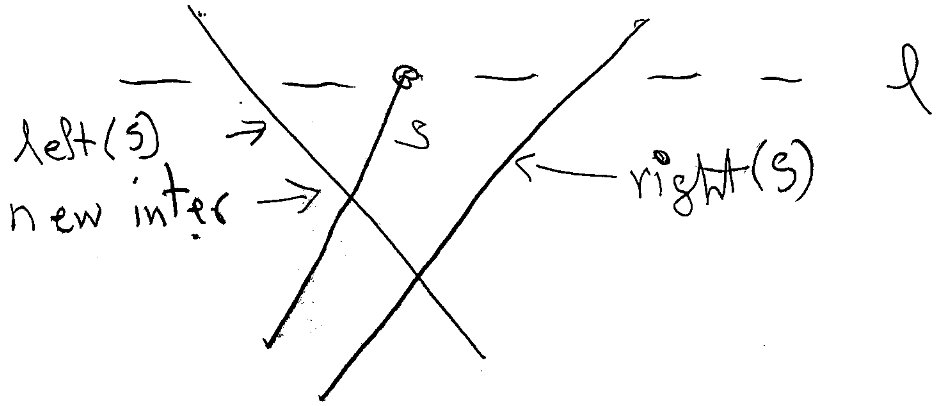
Proc HandleEvent(P)

	Cost	#
1) If P is upperend of S then		
a) insert(S, D)	$\log n$	n
b) add-inter(left(S), S, Q)	$\log n$	n
c) add-inter(S, right(S), Q)	$\log n$	n
2) If P is lowerend of S then		
a) add-inter(left(S), right(S), Q)	$\log n$	n
b) delete(S, D)	$\log n$	n
3) if $P = S \cap S'$ and $S < S'$		
a) swap(S, S', D)	$\log n$	I
b) add-inter(left(S'), S', Q)	$\log n$	I
c) add-inter(S, right(S), Q)	$\log n$	I
d) Report ;		

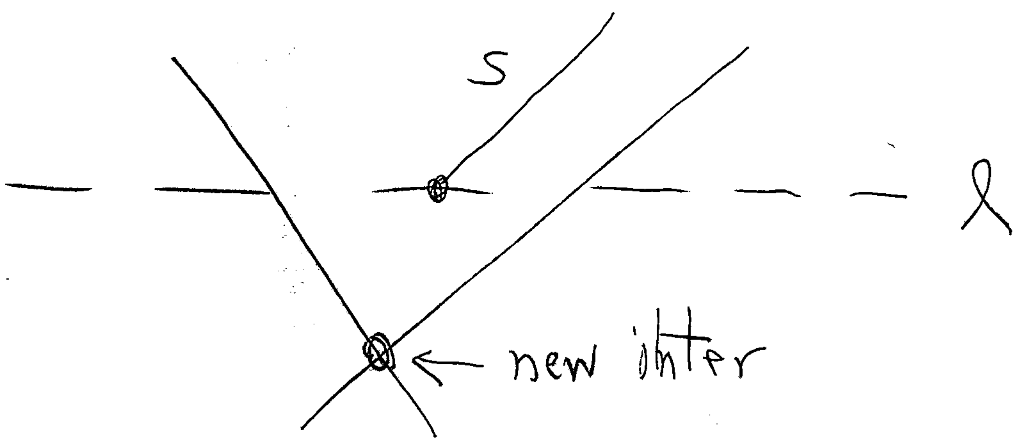
$O((n+I)\log n)$

Examples of events

1)



2)



3)