Computational Geometry
Intro

15-750
2/15/16

We have been in 1D!

eg sorting, searching, priority queues

These lectures move to 2D.

Apps (even for 2D)	More dim
Graphics	3d Sci Comp
Robotics	Machine Learning
Geo info systems	d very large
CAD/CAM	
Sci Comp	

Basic approach

Build complex obj out of simple objs.

eg Image ≡ array of dots

Integrated Circuit ≡ planar tris

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

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

Using points to generate objs

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

Linear Combinations

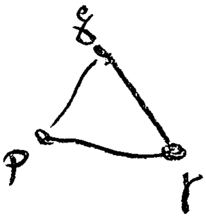
Subspace $= \sum \alpha_i P_i$ for $\alpha_i \in \mathbb{R}$

Affine Comb

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

Convex Comb

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

eg 

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

Primative Ops.

1) Equality $P = q$?

2) Line seg intersection test

3) Line side test

Inputs: (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 $L_1 = [P_1, P_2]$ & $L_2 = [P_3, P_4]$

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

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

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

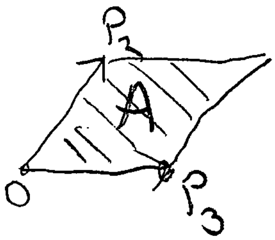
solve
$$\overset{\displaystyle A}{\overset{\shortparallel}{\begin{pmatrix} x_1 & x_2 & -x_3 & -x_4 \\ y_1 & y_2 & -y_3 & -y_4 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}}}\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad \Big| \quad \det(A) \neq 0$$

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

---

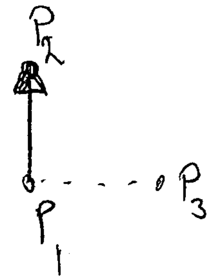
3) Line side test

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 $P_1 \to P_2$



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

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

$P_2' = P_2 - P_1 \qquad P_3' = P_3 - P_1$

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

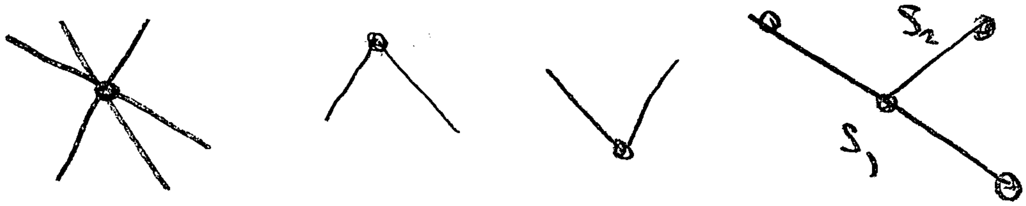
Optimal: Random Incremental

Sweep Line Alg

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

Assume: 1) no horizontal segs

2) cases not handled today!



≥ 3 seg at a point

Let $P \equiv$ Seg endpoints

$I \equiv$ Seg intersections

Events $\equiv P \cup I$

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

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

Note Order only changes at $P \cup I$

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 Insert P into Q

While $Q \neq \emptyset$

P = ExtractMax(Q)

HandleEvent(P)

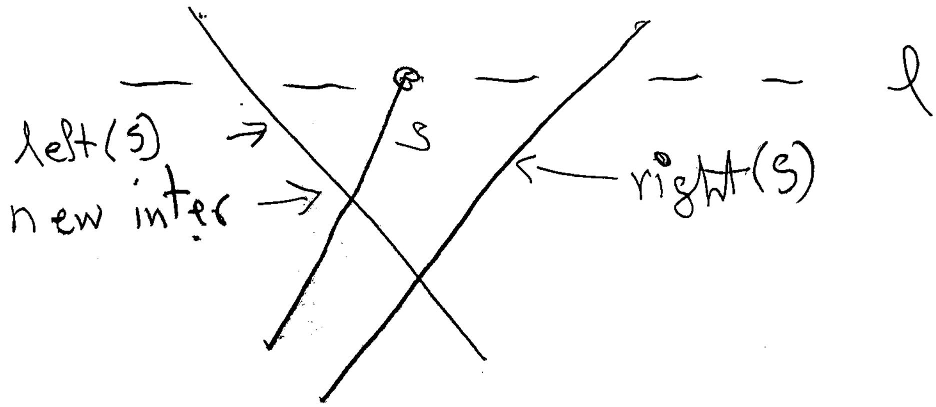
HandleEvent(P)

	cost	#
1) if P is upperend of S then		
insert(S, D)	$\log n$	n
add-inter(left(S), S, Q)	$\log n$	n
add-inter(S, right(S), Q)	$\log n$	n
2) if P is lowerend of S then		
add-inter(left(S), right(S), Q)	$\log n$	n
delete(S, D)	$\log n$	n
3) if $P \in S \cap S'$ & $S < S'$		
swap(S, S', D)	$\log n$	I
add-inter(left(S'), S', Q)	$\log n$	I
add-inter(S, right(S), Q)	$\log n$	I
report P		

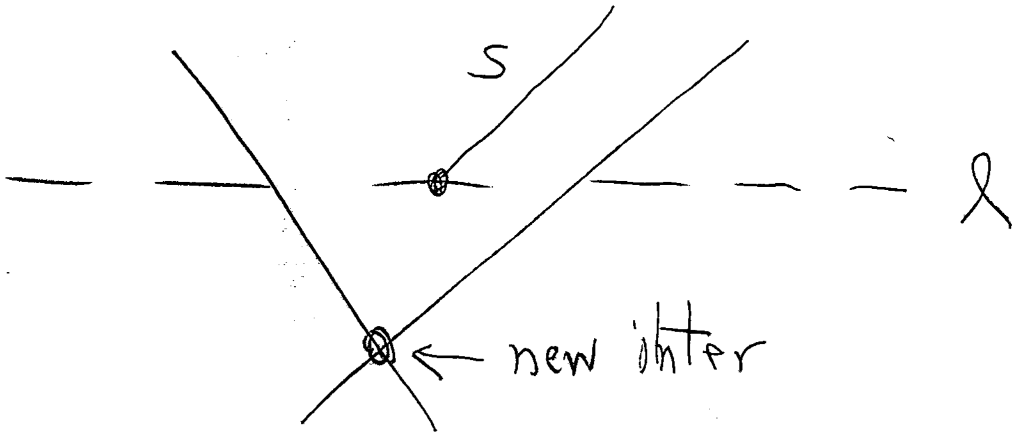
$O((n + |I|)\log n)$

Examples of events

1)



2)



3)