Computational Geometry
Intro

15-852
9/16/12

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 ≡ array of dots

Integrated Circuit ≡ planar trees

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 Segment	Pair of Points
Triangle	Triple of Points

Using points to generate objs

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

Linear Combinations

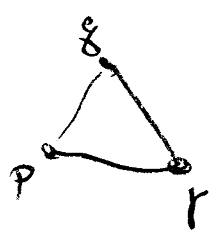
Subspace $= \sum \alpha_i P_i$ for $\alpha_i \in \mathbb{R}$

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 

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

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

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

st $\alpha_1 + \alpha_2 = 1$

$\alpha_3 + \alpha_4 = 1$

$\alpha_1, \ldots, \alpha_4 \geq 0$

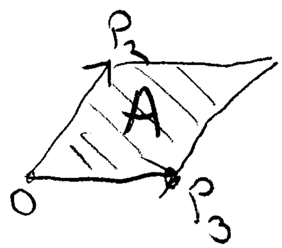
solve
$$\begin{pmatrix} x_1 & x_2 & -x_3 & -x_4 \\ y_1 & y_2 & -y_3 & -y_4 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

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

---

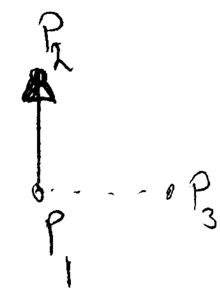
3) Line side test

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 $P_1 \to P_2$



<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

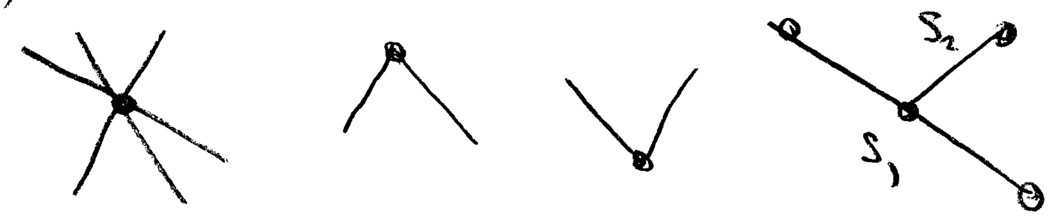
Optimal: Random Incremental

Sweep Line Alg

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

Assume: 1) no horizontal segs

2) cases not handled



≥ 3 seg at a point

Let $P \equiv$ Seg end points

$I \equiv$ 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 order 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.

<u>Alg</u> Insert P into Q

While $Q \neq \emptyset$

  P = ExtractMax(Q)

  HandleEvent(P)

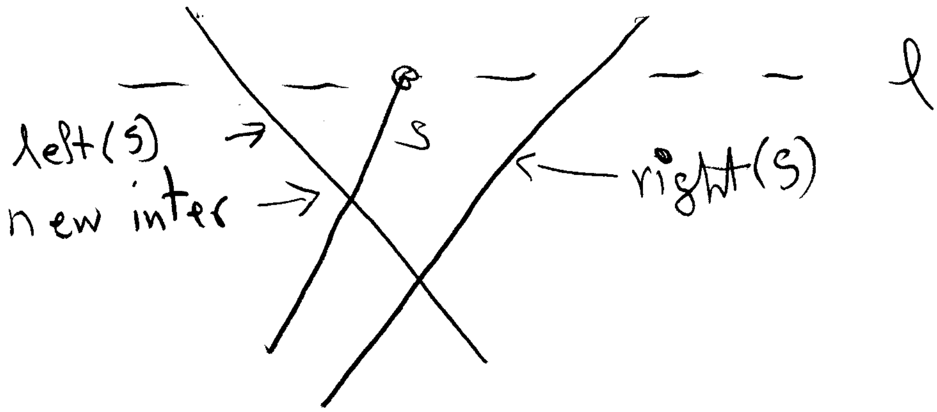
HandleEvent(P)

	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 $P \in S \cap S'$		
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		

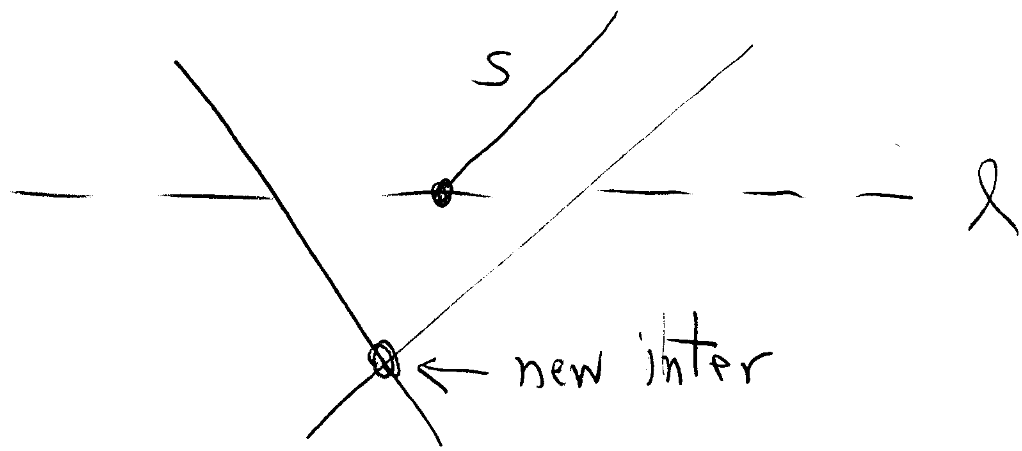
$O((n+|I|)\log n)$

Examples of events

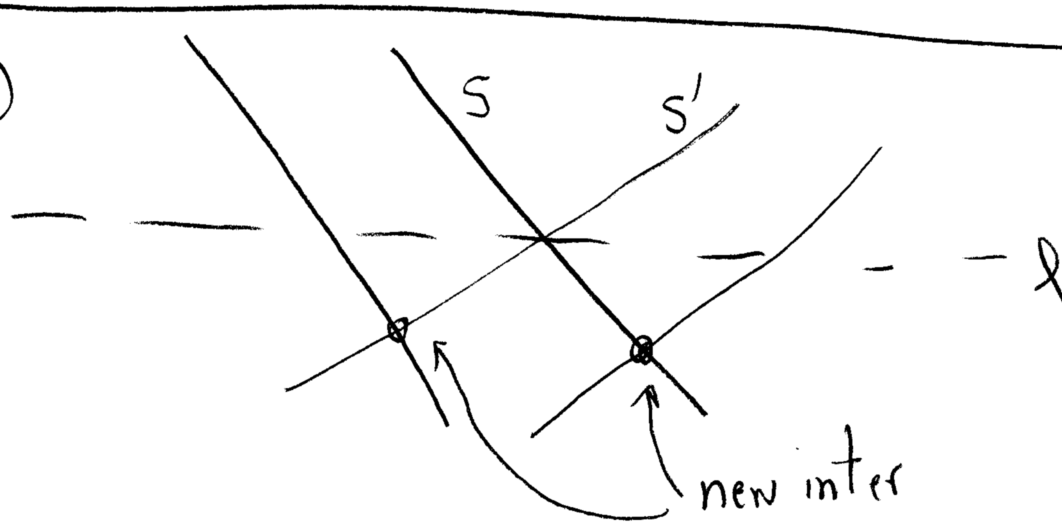
1)



2)



3)



Map Overlay Prob

Input: Segments $S = \{s_1, \dots, s_n\}$ (no hori seg)

Output: Break all seg in subseg s.t.
2 subseg intersect only at endpoints

Alg: SweepLine

Events: All seg intersections, I.
+ All end-points, P

Handling Events

Linked Lists:

$\mathcal{U}(P)$ = Subseq with upper end point P.

$L(P)$ = " "lower" "

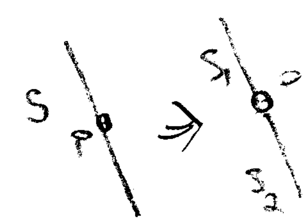
$C(P)$ = " " intersection in P.

Init $\mathcal{U}$ & L using S.

Init: $C(P) = \emptyset$

Procedure: HandleEvent(P point; T tree; Q queue)

0) $\forall s \in C(P)$: Form new subseq S_1, S_2
add S_1, S_2 to $\mathcal{U}(P)$ & $L(P)$



1) $\forall s \in L(P)$: delete(s, T)

2) $\forall s \in \mathcal{U}(P)$: insert$(s, T)$

3) $\forall$ new-neigh-pairs add inter to Q

Runtime Analysis

Let $m = \#$ subsegs output. $m \geq n$, $m \geq |I|$

<u>Claim</u> Alg runs in $O(m \log n)$ time.

pf $\exists$ at most m delete/inserts into T & Q.

How many Sub segs?

To show: $\# \text{subsegs} = O(n + |I|)$

Embedded Planar Graph $(G = (V, E), \varphi : G \to \mathbb{R}^2)$

$\varphi(e) = \text{path}$

$\varphi(e) \cap \varphi(e') = \text{only endpoints}$

Eulers Formula (G, φ)

$n_v = \#$ vertices

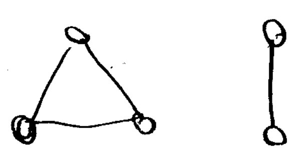
$n_e = \#$ edges

$n_f = \#$ connected boundaries (faces)

$c = \#$ connected components

then $n_f - n_e + n_v = 2c$

eg



$n_v = 5$

$n_e = 4$

$n_f = 3$ (why?)

$c = 2$

$5 - 4 + 3 = 2 \cdot 2 = 4$

Claim $3n_v \geq n_e$

pf 1) add edges until G is connected

2) " " each face size is 3.

3) no parallel edges.

$3n_f \leq 2n_e \qquad n$

$n_f \leq 2/3\, n_e$

$2/3\, n_e - n_e + n_v \geq 2$

$-1/3\, n_e + n_v \geq 2$

$n_v \geq 2 + 1/3\, n_e$

$3n_v > n_e$

Sweep Line is $O((n+I)\log n)$ time

Wrong! We sorted the intersection!