Triangulating a Polygon

CG
9/23/08

Recall: PSLG ≡ Planar Straightline graph.

Def (Simple) Polygonal chain is a PSLG consisting of a simple cycle P.

Claim A Polygonal chain has a unique interior.

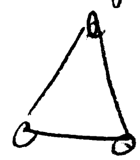
Def Polygon is Polygonal chain + interior

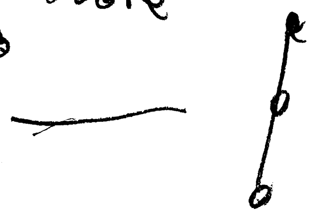
Triangulation: Addition of seg so that

1) Still PSLG
2) Interior is decomposed into triangles

Thm Every simple Polygon can be triangulated.

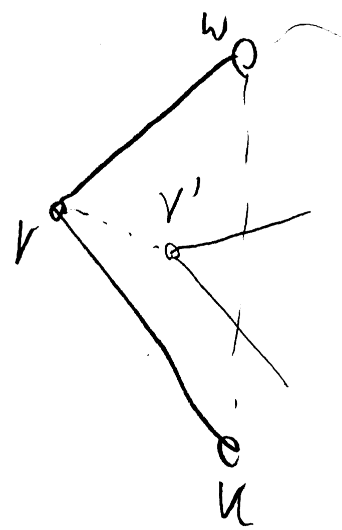
Pf Induct on # of seg or points n

base case: $n=3$  done

Assume no 180° angles 

$n>3$ true for $m<n$

Let v be left most point with neigs w & u



1) Case 1 seg $= [w,u]$ is interior

We get two Poly $|P_1|=3$ $|P_2| = n-1$

2) Case 2 $\exists$ point v' interior to Tri $= [w,v,u]$

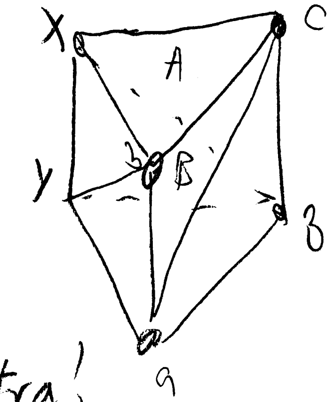
Let v' be left most such point

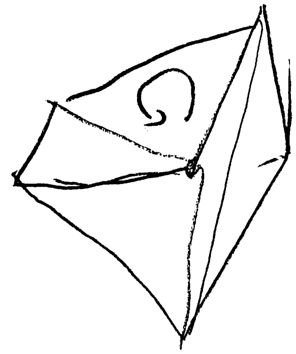
Seg $= [v,v']$ is inter

$|P_1| < n$ & $|P_2| < n$

Thm Not Every simple polygonal surface (Polytope) in 3D can be decomposed in Tetrahedra

eg prism





By Contra!

Consider Tet with face B

Missing vertex is X or Y not Z

not X since seg [X, a] outside,

" Y " [Y, C] "

In general prob is NP-Hard

Guarding A Polygon

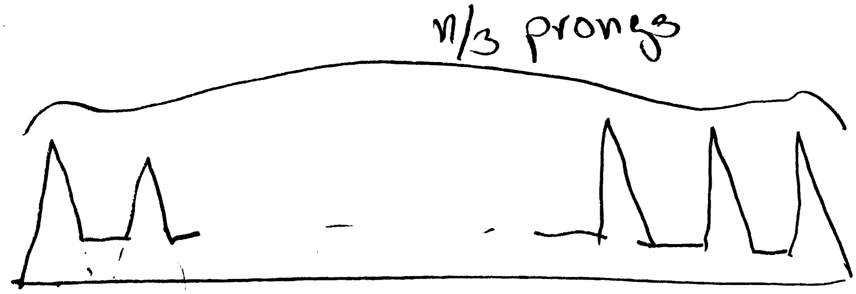
Input: Polygon P

Output: locations $p_1 \cdots p_k \in P$ (guards)

1) Guards cover P
2) k small.

Thm A polygon P with n vertices $\lfloor n/3 \rfloor$ guards suffice and maybe necessary.

Necessary: P =



$|P| = n$ Needs a guard per prong.

$n/3$-guards Alg(P)

1) Tri P $\bar{P}$
2) 3-color $\bar{P}$
   a) Construct geometric dual T (a tree)
   b) 3-Color $\bar{P}$ by traversing tris in an inorder fashion.
3) Pick least used color.

Only non-linear time step is 1)

2D-algorithm

Proof $\Rightarrow O(n^2)$

Known: $O(n)$ Chazelle

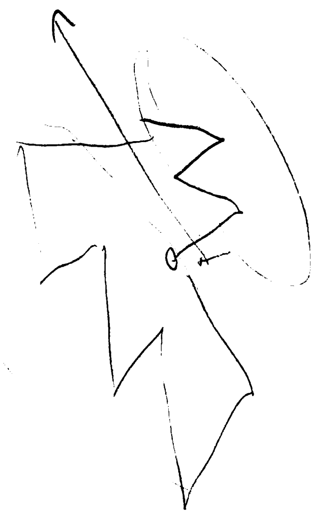
Today: $O(n \log n)$ [illegible] (sweep line)

This Class: $O(n \log^* n)$ Seidel (incremental randomized)

Def $\log^* n = \min_k \underbrace{\log \log \cdots \log}_{k} n \leq 1$

Prob: Give a $O(n)$ time alg to determine which side of an edge is interior/exterior (of a simple poly)

Prob: test $p \in Int(P)$ in $O(n)$ time.



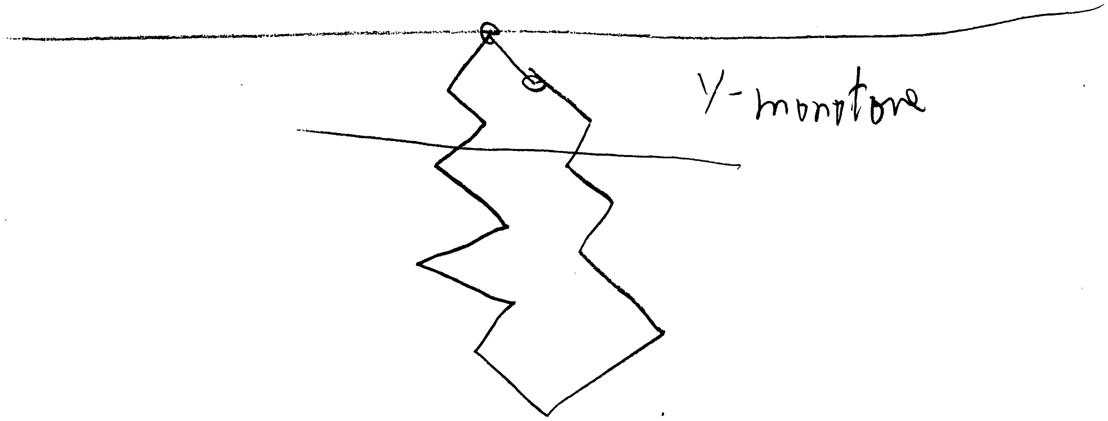
3.14 $O(n \log n)$ OK

$O(n)$?

Trap $\Rightarrow$ Tri

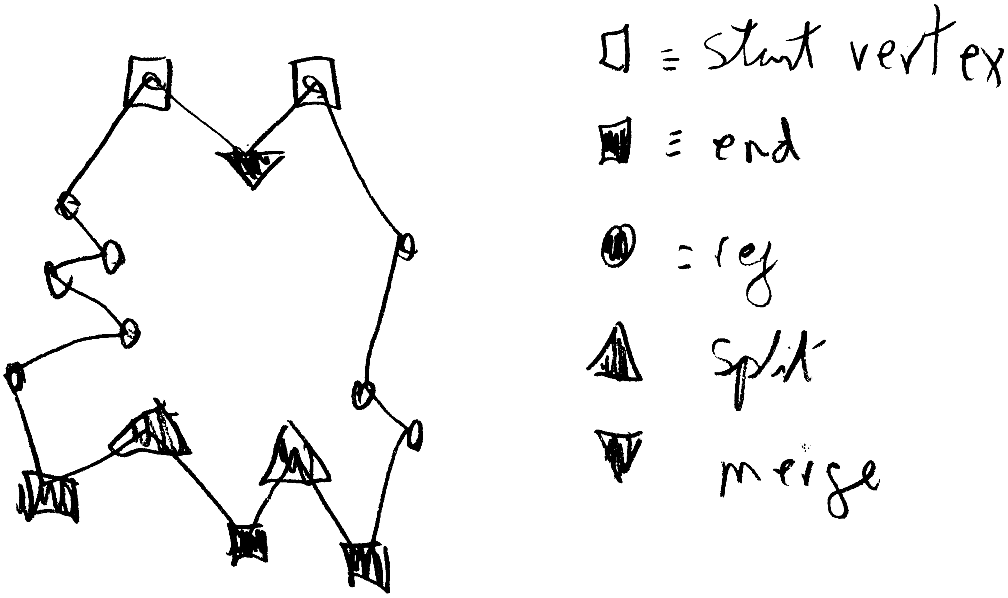
Step 1 Partition into Monotone Polygons

Def P is Y-monotone if
Every horizontal line l
$l \cap P$ is connected or empty



Alg type: Line Sweep $O(n \lg n)$ time

Def



Claim 3.4 P is y-monotone iff no split or merge vertices

(⇒) easy

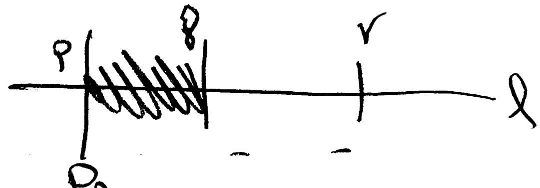
(⇐)

not mono ⇒ ∃ split or merge

∃ l with at least 2 intervals

∃ l' with one — false

⇒ ∃ l'' going from 2 to 1

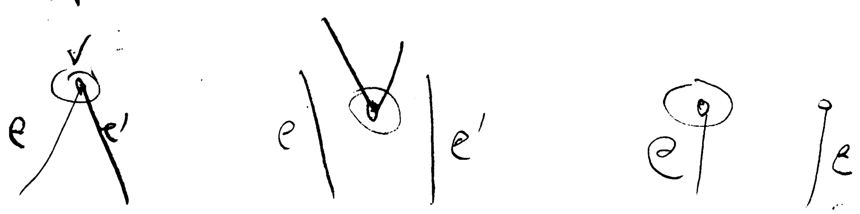


∃ path from q to r disjoint from p

Alg Line Sweep

1) Events are endpoints
2) Dictionary
    1) Even # of segments in pairs (intervals)
    2) each IS has a helper vertex

$\text{helper}_l(e, e')$ = lowest vertex above l and between e & e'



3) Degeneracy: no horizontal seg.

Make Monotone (q ! event)

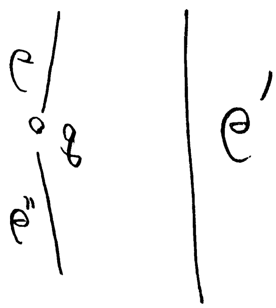
Case (Start Vertex)

1) Add new interval
2) set helper $\Leftarrow q_e$

Case (End Vertex)

1) if helper is a merge vertex then add (q, helper)
2) remove interval & helper.

Case (Regular)



1) add (q, helper)
2) replace e with e''
3) helper $\Leftarrow$ q

Case (Split)

1) add (q, helper)
2) "split" interval
3) helps $\Leftarrow$ q

Case (Merge)

1) add($help_{PL}$, 8) add($help_{PR}$, 8)

2) "Merge" intervals

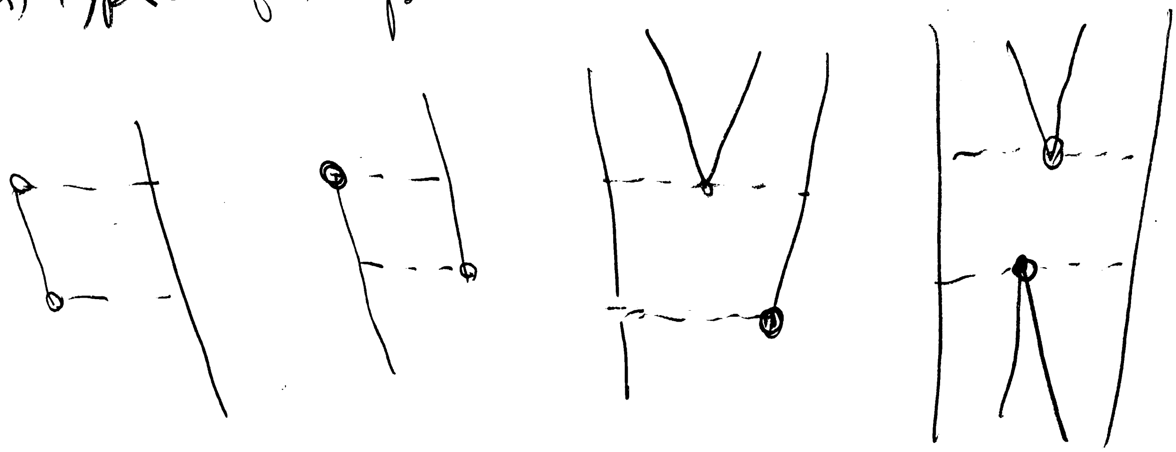
3) help ← 8

Another View

1) Make Trapezoidal Decom (sweep line)

2) For each trap add a diagonal if possible

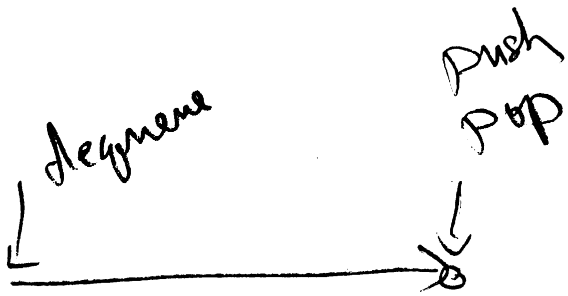
a) types of Traps



Triangulating a Monotone Poly

Y-monotone

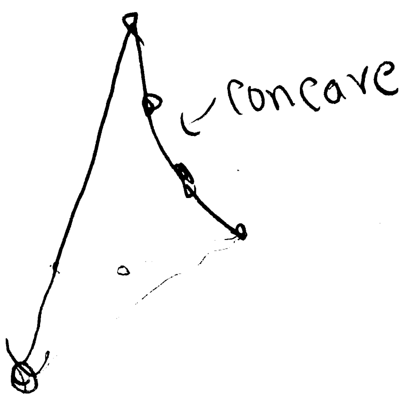
Q ≡ 2-sided queue
opts
1) push
2) pop
3) dequeue



Process points in decreasing Y-value.

Maintain

This or Reflection



WLOG

Event 8

Case (8 in right chain)

1) Push 8

2) Pop until chain is concave

Case (8 in left chain)(

1) Dequeue chain

$O(n)$

Each vertex push at most once.