Triangulating a Polygon

CG
9/23/10

Recall: PSLG ≡ Planar Straightline graph.

<u>Def</u> (Simple) <u>Polygonal chain</u> is a PSLG consisting of a simple cycle P.

<u>Claim</u> A Polygonal chain has a unique interior.

<u>Def</u> <u>Polygon</u> is Polygonal chain + interior

Triangulation: Addition of seg so that

1) Still PSLG
2) Interior is decomposed into triangles

<u>Thm</u> Every simple Polygon can be triangulated.

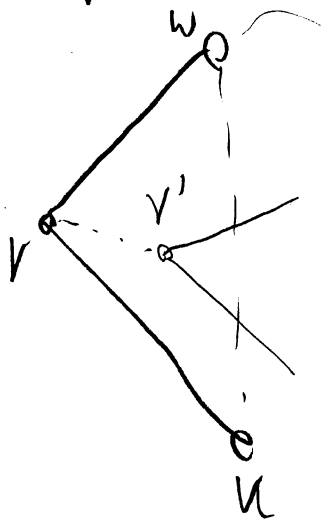
Pf Induct on # of seg or points n

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' inter to Tri $=[w,v,u]$

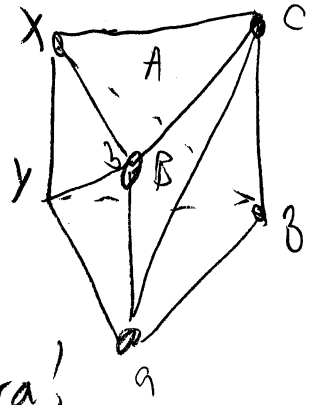
Let v' be left most such point

Seg $\equiv [v,v']$ is inter

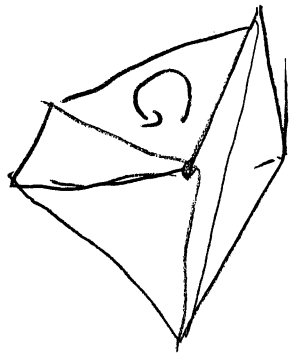
$|P_1|<n$ & $|P_2|<n$

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

eg <u>prism</u>



twist Top



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

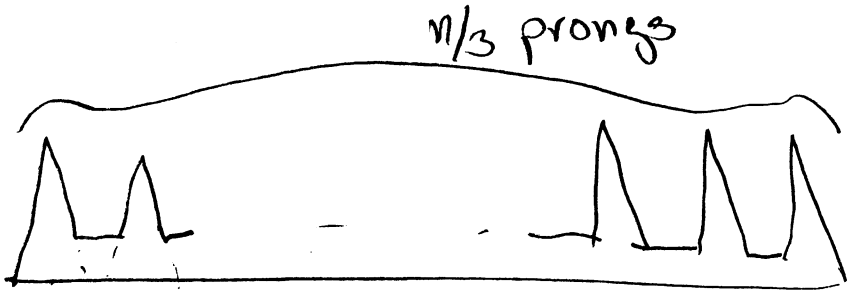
Input: Polygon P

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

1) Guards cover P
2) k small.

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

Necessary:

$P =$ 

$|P| = \Omega$ 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

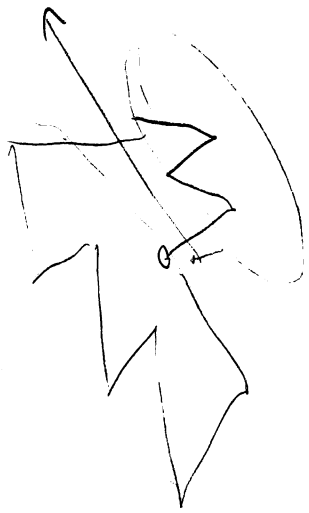
Today: $O(n \log n)$ Fine (sweep line)

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

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

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.



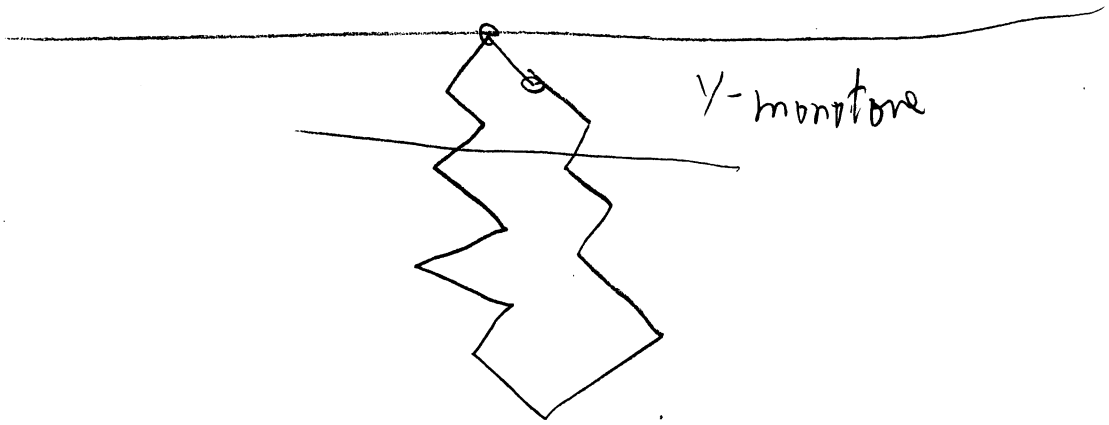
3.14 $O(n \log n)$ OK

$O(n)$?

Trap $\Rightarrow$ Tri'

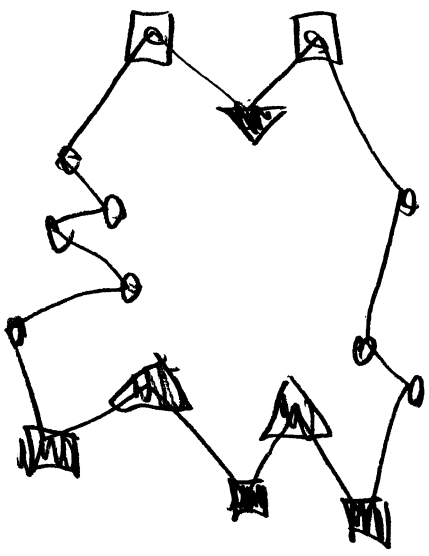
Step 1 Partition into Monotone Polygons

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



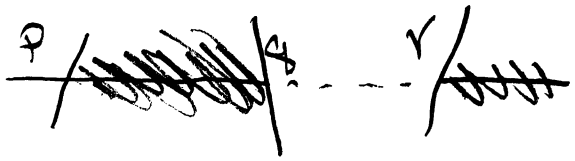
□ ≡ start vertex
■ ≡ end
● = reg
▲ split
▼ merge

<u>Claim</u> P is y-monotone iff no split or merge vertices.

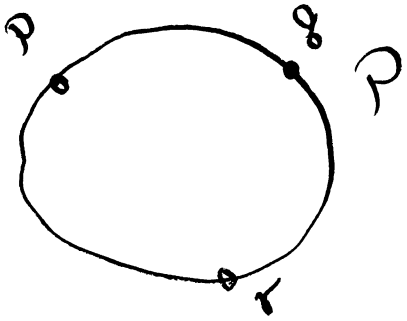
($\Rightarrow$) (easy)

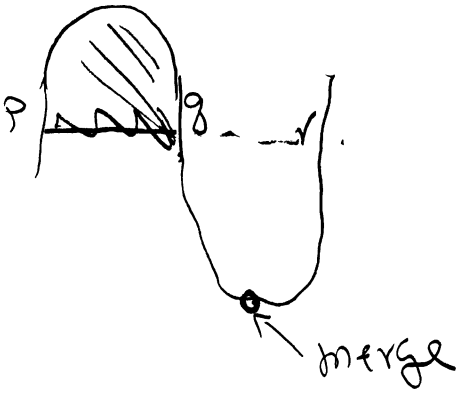
($\Leftarrow$) (not mono $\Rightarrow$ $\exists$ split or merge)

Assume not mono

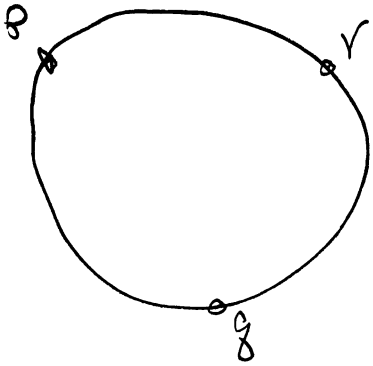


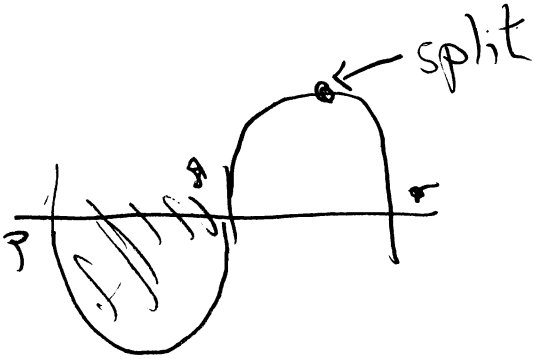
<u>Case 1</u>





<u>Case 2</u>





Alg Sweep Line (top-to-bottom)

Events: endpoints

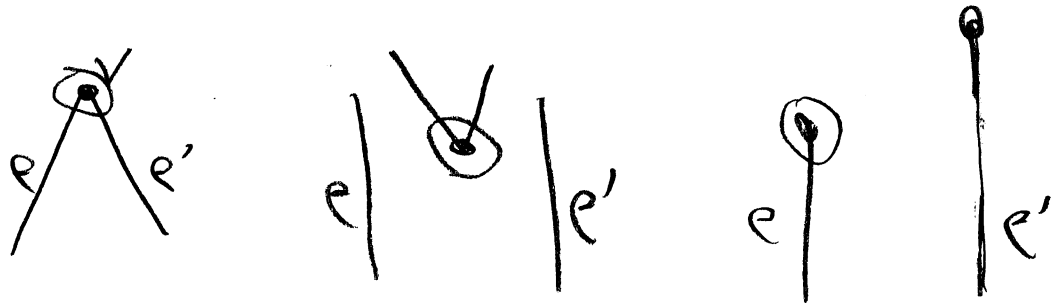
Dictionary: Intervals (sorted)

Interval: (left-seg, right-seg, helper vertex)

vertex: 2 edges before & after

Def helper(e, e') ≡ lowest vertex above ℓ and between e & e'.

eg



no horizontal seg.

Procedure add(p, q) ≡ add seg from p to q.

Make Monotone(q : event)

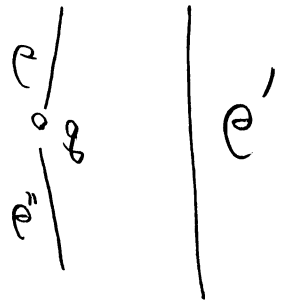
Case (Start Vertex)

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.

Case (Regular)



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

Case (Split)

1) add (q, helper)
2) "split" interval say I_1, I_2
3) helper(I_1) = helper(I_2) $\leftarrow q$

Case (Merge)

1) add($help_L$, q); add($help_R$, q)

2) "Merge" intervals

3) help ← q

Another View

1) Make Trapezoidal Decom (sweep line)
2) For each trap add a diagonal if possible
   a) types of Traps