15-456
1/28/13

Triangulating a Polygon

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.

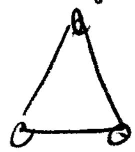
Def Polygon is Polygonal chain + interior

Triangulation: Addition of seg so that

1) Still PSLG
2) Interior is decomposed into triangles

Thm Every simple Polygon can be triangulated.

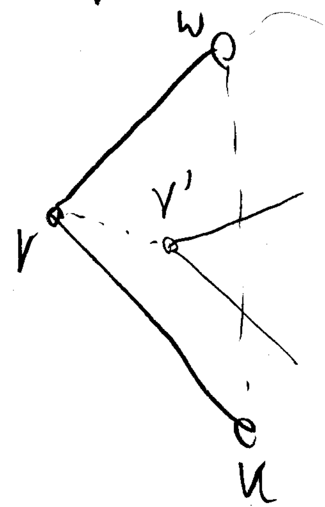
Pf Induct on # of seg or points n

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

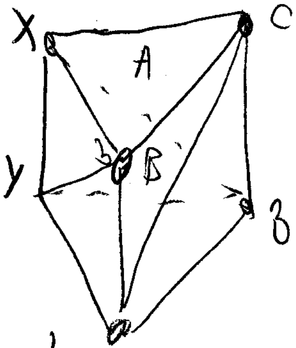
Let v' be left most such point

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

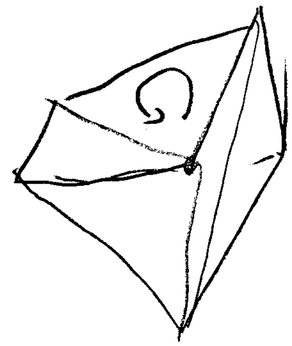
$|P_1| < n$ & $|P_2| < n$

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

eg prism



twist Top



By Contra!
Consider Tet with face B

Missing vertex is X or Y not z
not X since seg [a, X] is outside.
In not Y since seg [z, Y] is outside.

In General: Test if polygonal surface is decomposible is NP-Hard.

Guarding A Polygon

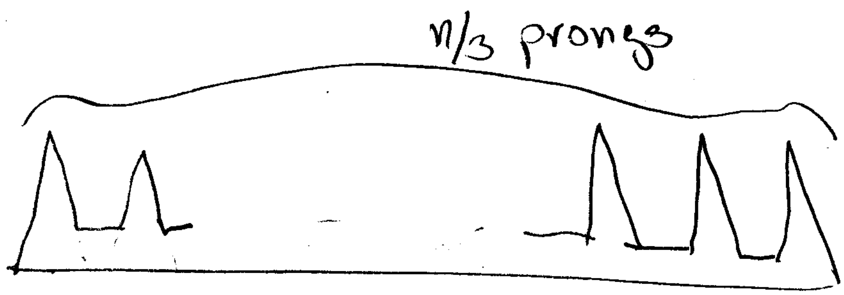
Input: Polygon P

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 degree 3 tree)
   b) 3-Color $\bar{P}$ by traversing tri's 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

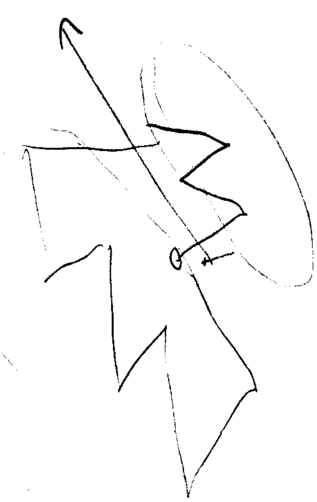
Today: $O(n \log n)$ (sweep line)

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 is $Int(P)$ in $O(n)$ time.



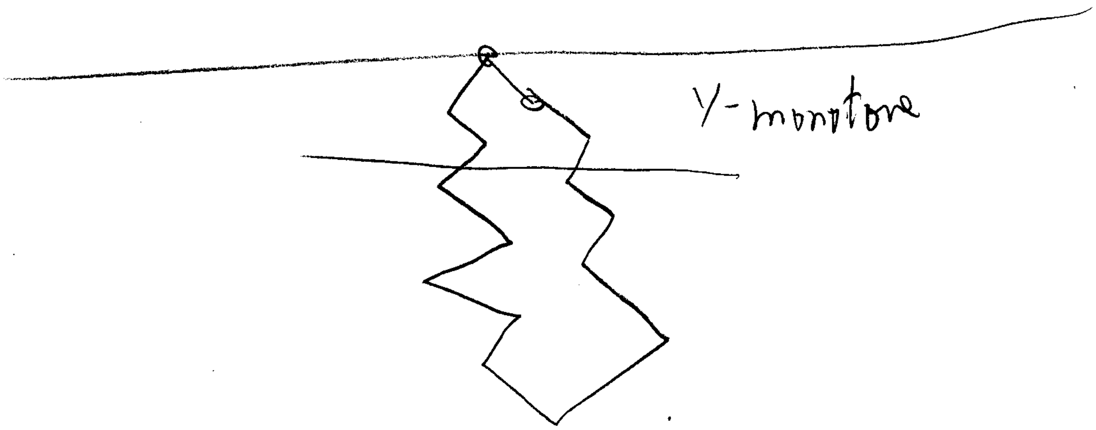
3.14 $O(n \log n)$ OK

$O(n)$?

Trap $\Rightarrow$ Tri

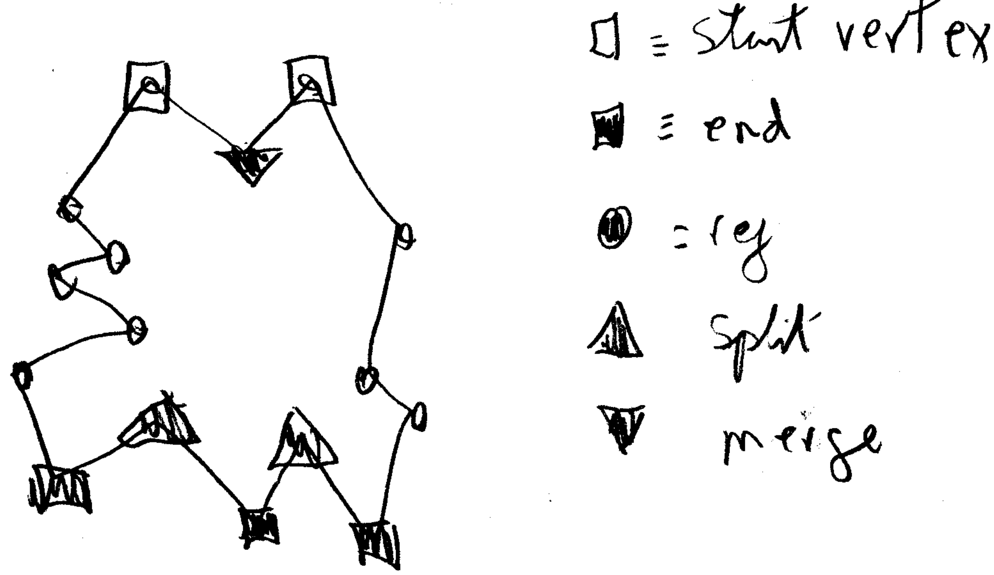
Step 1 Partition into Monotone Polygons

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

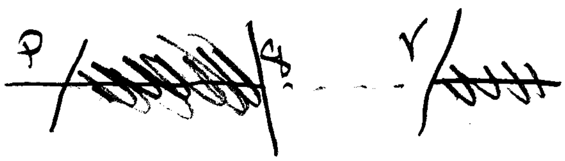


<u>Claim</u> P is y-monotone iff no split or merge vertices.

$(\Rightarrow)$ (easy)

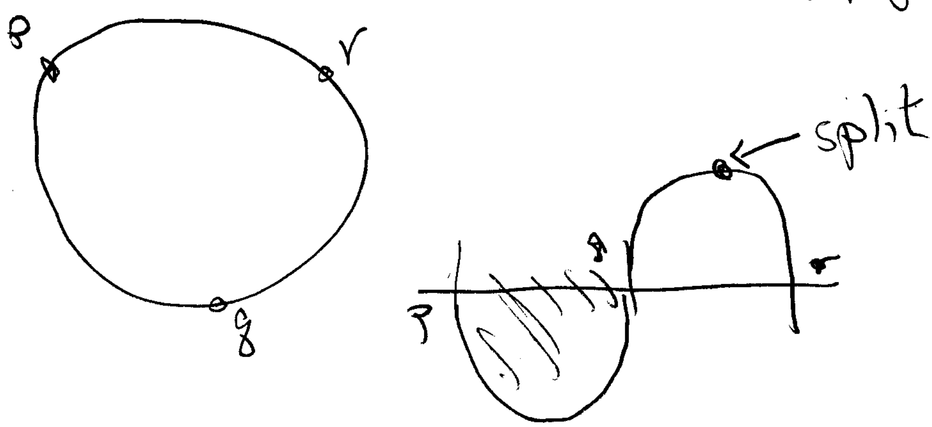
$(\Leftarrow)$ (not mono $\Rightarrow \exists$ split or merge)

assume not mono



<u>Case 1</u>

<u>Case 2</u>



Alg Sweep Line (top-to-bottom)

- Events: endpoints

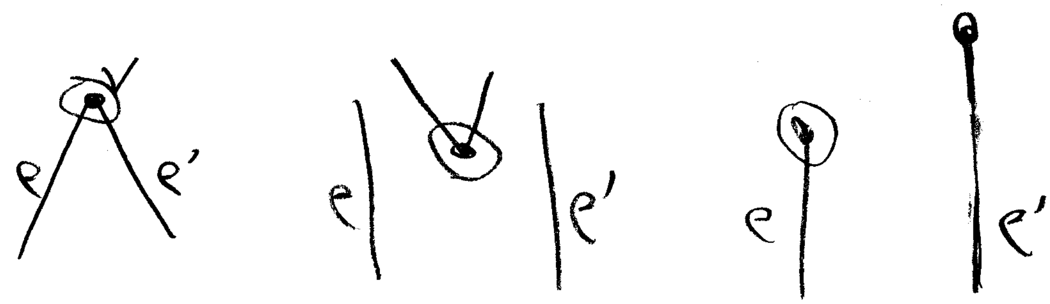
Dictionary: Intervals (sorted)

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 horizonal seg.

Procedure add(p, q) = add seg from p to q if [p, q] not already an edge.

Make Monotone (q : event)

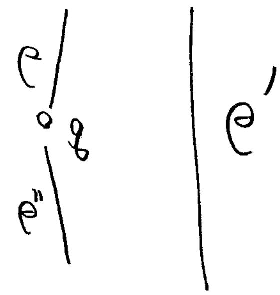
Case (Start Vertex)

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

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($help_{PL}$, g); add($help_{PR}$, g)

2) "Merge" intervals

3) help $\leftarrow g$

Another View

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

a) types of Traps