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
Then every simple polygon can be triangulated.

1. Induct on # of seg on points N

Base case: $n=3$ \hspace{1cm} done

Assume no 180° angles

$n > 3$ True for $m < n$

Let $v$ be left most point with neighbors $w$ & $u$

1. Case 1. $\deg = \{w, u\}$ is interior

We get two Poly $P_1 = 3$ \hspace{1cm} $P_2 = n-1$

2. Case 2. 3 point $v'$ interior to $\text{Tri} = \{v, v', w\}$

Let $v'$ be left most such point

$\deg = \{v, v'\}$ in interior

$r_1 
\leq n \leq r_2 \leq n$
Thm Not Every Simple polygonal surface in 3D can be decomposed in Tetrahedra

Ex: Prism X twist top

By Contradiction Consider Tet with faces B

Missing vertex in X or Y not Z

Not X since seg [a, x] is outside O

Not Y since seg [a, y] is outside O

In General: Test if polygonal surface is decomposable in NP-Hard
Guarding A Polygon

Input: Polygon $P$

Output: locations $p_i, i \in P$ (guards)

1) Guards cover $P$
2) $k$ small.

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

Necessary:

$P = \begin{array}{c}
\text{n/3 prongs}
\end{array}$

$|P| = n$ Needs a guard per prong.
$\frac{7}{3}$-guards Alg $(P)$

1) Tri $P$ $\overline{P}$
2) 3-color $\overline{P}$
   a) Construct geometric dual $T$ (a degree 3 tree)
   b) 3-color $\overline{P}$ 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: \( O(n \log^* n) \) (line sweep line)

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

\[
\text{Def: } \log^* n = \min_k \log \log \ldots \log n \leq 1
\]

Probs:
- Give a \( O(n) \) time alg to determine which side of an edge in interior/exterior of a simple poly.
- Test \( P \in \text{Int}(P) \) in \( O(1) \) time.

3.14 \( O(n \log n) \) OK

\( O(n) ? \)

\( \text{Triv } \Rightarrow \text{Triv}^1 \)
Step 1: Partition into Monotone Polygons

Def. P is monotone if
Every horizontal line l
in P is connected or empty

Alg. Type: line sweep
O(n log n) time

Def

- □ = start vertex
- ■ = end
- ○ = seg
- ▲ = split
- ▼ = merge
Claim $P \in \gamma$-monotone iff no split or merge vertices.

($\Rightarrow$) (easy)

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

Assume not mono

Case 1

Case 2
Alg: Sweep Line (top-to-bottom)

Events: endpoints

Dictionary: Intervals (sorted)

Interval: (leftseg, rightseg, helper vertex)

vertex: 2 edges before, after

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

no horizontal seg.

Procedure: add (p, q) = add seg from p to q if [p, q] not already an edge.
Make Monotone (e : event)

Case (Start Vertex)
1) Add new interval
2) set helper = e

Case (End Vertex)
1) if helper is a merge vertex then add (g, helper)
2) remove interval

Case (Regular)
1) add (g, helper)
2) replace e with e''
3) helper <= g

Case (Split)
1) add (g, helper)
2) "split" interval say I₁, I₂
3) helper(I₁) = helper(I₂) <= g
Case (Merge)

1) add (helpL, 8); add (helpR, 8)

2) "Merge" intervals

3) help <= 8
Another View

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

\[ \text{Diagrams showing different types of traps.} \]