L2

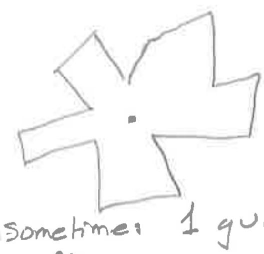
Guarding an Art Gallery (n vertex simple polygon P)

every point p in P should be "visible" from a guard

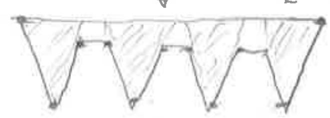
$g \in P$

$\lfloor n/3 \rfloor$ guards = points inside P are sufficient and sometimes necessary

[Chvatal 1975]



(sometimes 1 guard suffices)



$n=12$

at least 4 guards

Fisk's Proof (from "The Book")

<See notes from previous class>

What if we want the minimum number of guards? or even to decide if k guards are enough

NP-hard — meaning if we could solve in polytime then ALL problems in NP could be solved in polytime.

[NP is a collection of decision problems that no one knows how to decide in polytime but they can be decided in polytime given a polynomial-sized hint.]

they're not ridiculously difficult (like Halting Problem)

[but not believed to be in NP since it is also hard for $\exists\mathbb{R}$ existential theory of the reals]

$P \subseteq NP \subseteq \exists\mathbb{R} \subseteq PSPACE$ $\exists x_1 \dots x_n \; f(x_1 \dots x_n)$, $x_i \in \mathbb{R}$

Approximation algorithm $O(\lg\lg \underbrace{OPT}_{\text{min \#guards needed}})$-approximation algorithm

Kirkpatrick 2015

Algorithm to Δate simple polygon

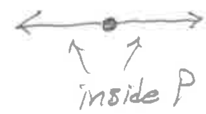
Our proof gave $O(n^2)$ time algorithm.

Chazelle [1991] gave $O(n)$ algorithm

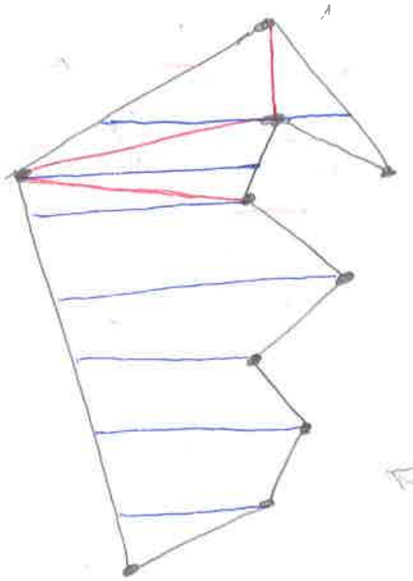
We'll discuss $O(n \log n)$ algorithm based on trapezoidization
[most use ↑]

$O(n \log n)$ ① trapezoidize simple polygon

cut polygon horizontally from each vertex



Assume no two vertices have same y-coord.



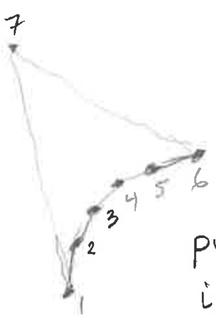
all vertices on right in this piece

Every trap in same piece has vertices on same side

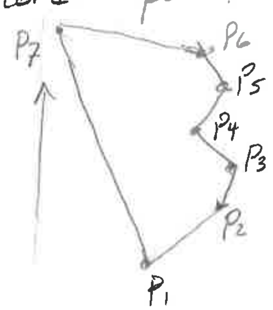
why?

$O(n)$ ② Compute Δtion from ⏢tion

Every ⏢ has one vertex on top and one on bottom
Connect them by chord (if not already an edge of P)
These chords partition P into unimonotone polygons (y monotone curve plus one edge)



push p_1 & p_2 on stack S
$i=3$
while



Cut off an ear at convex vertex (not top or bottom)
repeat

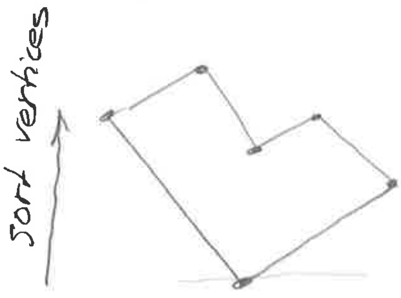
Step (1) Trapezoidize polygon

Use a sweep line algorithm:

sweep horizontal line from bottom to top

track changes to cross section

(intersection of sweep with poly)

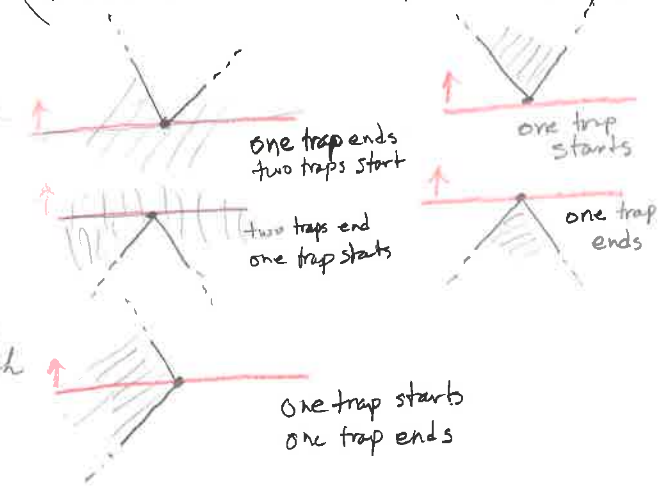


Events for simple polys

2 Edge birth

2 Edge death

1 Edge death & 1 edge birth



- Events occur in sorted order of points
- Use neighbors on sweep to determine ~~traps~~ zoids

key: Keep edge intersections in binary search tree ordered by x-coord.

Note: order never changes between events
<no intersections>

Events = n

time per event
$= O(\log n)$

initial sort by y
$= O(n \log n)$

$O(n \log n)$

What events for arrangement of segments that might intersect?