Guarding an Art Gallery

(n vertex simple polygon P)

\[ \lceil \frac{n}{3} \rceil \text{ guards are sufficient and sometimes necessary} \]

[Chvátal 1975]

\[ n = 12 \]

at least 4 guards

Fisk's Proof (from "The Book")

See notes from previous class

What if we want the minimum number of guards?

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

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

but not believed to be in NP since it is also hard for \( \exists R \), existential theory of the reals

\[ P \subseteq NP \subseteq \exists R \subseteq PSPACE \]

Approximation algorithm \( O(\log^{O(1)} \text{OPT}) \)-approximation algorithm

mix guards needed

Kirkpatrick 2015
Algorithm to \Delta ate 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

\( O(n \log n) \) 

1. trapezoidize simple polygon

Assume no two vertices have same y-coord.

cut polygon horizontally from each vertex

\( O(n) \) 

2. Compute \( \Delta \) tion from \( \Delta \) tion

Every \( \square \) 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 unimontone (y-monotone curve plus one edge

push \( P_i \) & \( P_j \) on stacks
while i \eqq 3

Repeat

Cut off an ear at convex vertex (not top or bottom)
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 occur in sorted order of points
- Use neighbors on sweep to determine traps/voids

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^2)
- Initial sort by y = O(N log N)

What events for arrangement of segments that might intersect?