TRIANGULATING A MONOTONE POLYGON
RULE: Join to whatever you can to the left. When done, move forward.

can't join to anything

\[\text{sweep}\]
notice co-vertical
Notice we need to know direction of monotonicity

Proof of correctness?
MONOTONE POLYGON TRIANGULATION PROOF

\[ \text{Y}_{\text{max}} \]

\[ \text{Y}_{\text{min}} \]
MONOTONE POLYGON TRIANGULATION

$Y_{\text{max}} Y_{\text{min}}$ line splits $Y_2$ & $Y_3$  \( \Rightarrow \) connect $Y_2 Y_3$
MONOTONE POLYGON TRIANGULATION

\( \text{Proof} \)

\( Y_{\text{max}} Y_{\text{min}} \) line splits \( Y_2 \) & \( Y_3 \) ? \( \rightarrow \) connect \( Y_2 Y_3 \)
else

\( Y_{\text{max}} \) sees \( Y_3 \) ? \( \rightarrow \) connect \( Y_{\text{max}} Y_3 \)
MONOTONE POLYGON TRIANGULATION

PROOF

\( Y_{\text{max}} Y_{\text{min}} \) line splits \( Y_2 \) & \( Y_3 \) ? \( \rightarrow \) connect \( Y_2 Y_3 \)

else

\( Y_{\text{max}} \) sees \( Y_3 \) ? \( \rightarrow \) connect \( Y_{\text{max}} Y_3 \)

else \( \) connect \( Y_2 Y_{\text{min}} \)
n-gon $\rightarrow$ cut ear: monotone $(n-1)$-gon.

INDUCTION
$n$-gon $\rightarrow$ cut ear: monotone $(n-1)$-gon

\[ \text{INDUCTION} \]
1. $n$-gon $\rightarrow$ cut ear: monotone $(n-1)$-gon.

2. Left

3. Right but eventual left

Induction
n-gon \rightarrow \text{cut ear: monotone (n-1)gon.}

\text{INDUCTION}

① left

③ right but eventual left
1. n-gon → cut ear: monotone (n-1)gon.

2. left

3. right but eventual left

4. Induction
n-gon → cut ear: monotone (n-1)gon.

INDUCTION

left

right but eventual left
n-gon → cut ear: monotone (n-1)gon.

INDUCTION

①

② left

③ right but eventual left before
n-gon \to \text{ cut ear: monotone (n-1)gon.}

\text{INDUCTION}

\text{right but eventual left before}

\text{right but no return before}

\text{same}
Testing a polygon for monotonicity

In a specific direction: easy → it's a local property

call it $\bar{x}$

don't go back till you hit $x_{\text{max}}$

start at $x_{\text{min}}$
Monotonicity in any direction?
Monotonicity in any direction?

Get legal directions for one vertex:

Update legal cones incrementally
(maintain intersection)
I claim test only reflex vertices why?
Test only reflex vertices

suppose a non-extreme convex vertex fails test

(not extreme)
Convex

fail perpendicular test

extreme convex
Test only reflex vertices

Suppose a non-extreme convex vertex fails test

Somewhere a reflex will also fail

(not extreme)

Convex

Fail perpendicular test

Extreme convex
Looks like the intersection of all the valid angles (one per vertex) does have a high complexity (number of angular components). That would lead to an $O(n\log n)$-time algorithm.

There is a linear time algorithm which I will find and go over in class.

Could be a project too.