One-Bend Drawings of Outerplanar Graphs Inside Simple Polygons

Patrizio Angelini
John Cabot University Rome

Philipp Kindermann
Universität Trier

Andre Löffler
Universität Würzburg

Lena Schlipf
Universität Tübingen

Antonios Symvonis
National Technical University of Athens

Graph Drawing 2021

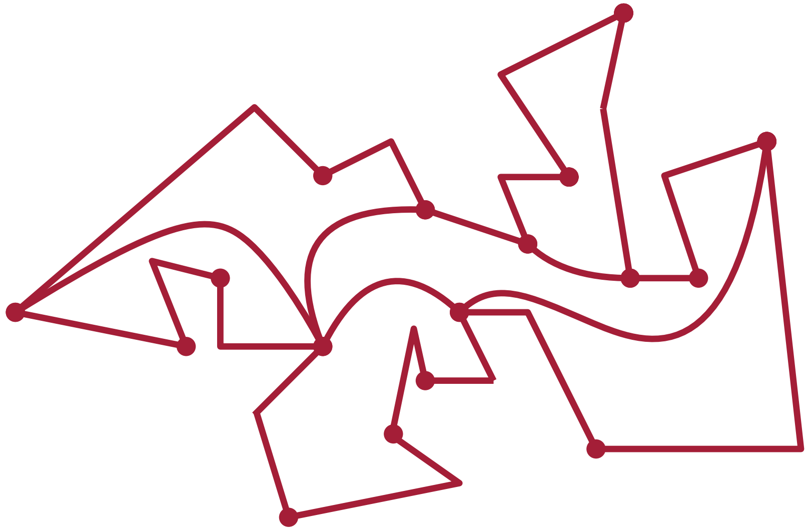
Problem

Input

- Biconnected outerplanar graph $G = (V, E)$
- Drawing of the outer face of G as simple polygon P

Question

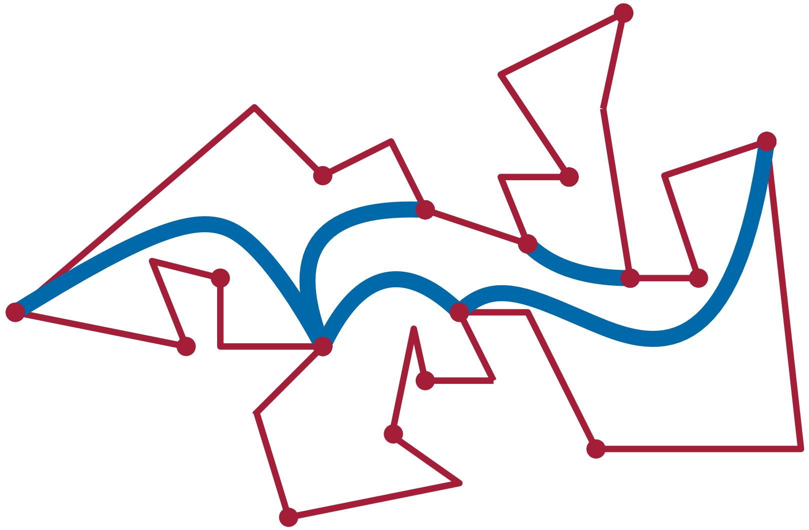
Does G admit an outerplanar drawing with at most one bend per edge that contains the given drawing of the outer face?



Problem

Input

- Biconnected outerplanar graph $G = (V, E)$
- Drawing of the outer face of G as simple polygon P



Question

Does G admit an outerplanar drawing with at most one bend per edge that contains the given drawing of the outer face?

Motivation & Related Work

- Prescribing the outer face: many results

[Tutte '63] [Hong, Nagamochi '07]

- Extension of straight-line drawings: NP-hard

[Patrignani '06]

- Allowing overlaps of edges and polygonal region: $\exists\mathbb{R}$-complete

[Lubiw, Miltzow, Mondal '18]

Non-straight edges:

- Extension of planar topological drawing: linear time

[Angelini et al '15]

- Linear number of bends per edge suffices (72 $|P|$)

[Chan et al. '15]

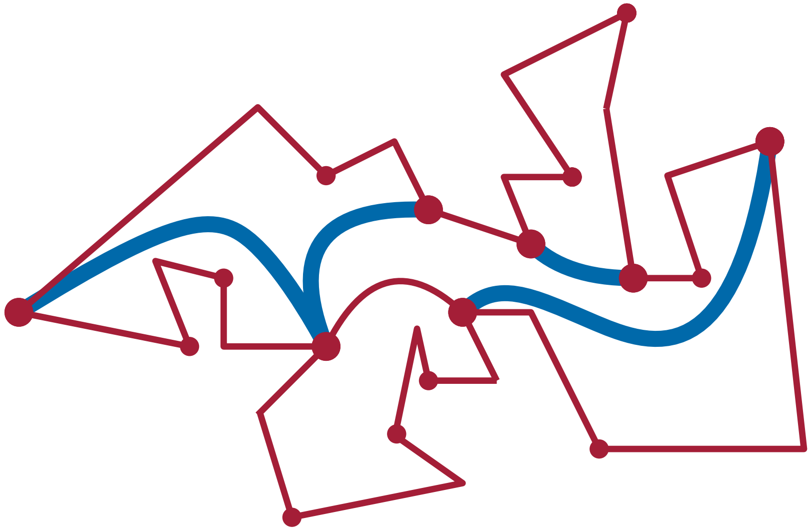
- P star-shaped: one bend suffices

[Mchedlize, Urhausen '18]

Problem

Input

- Biconnected outerplanar graph $G = (V, E)$
- Drawing of the outer face of G as simple polygon P



Question

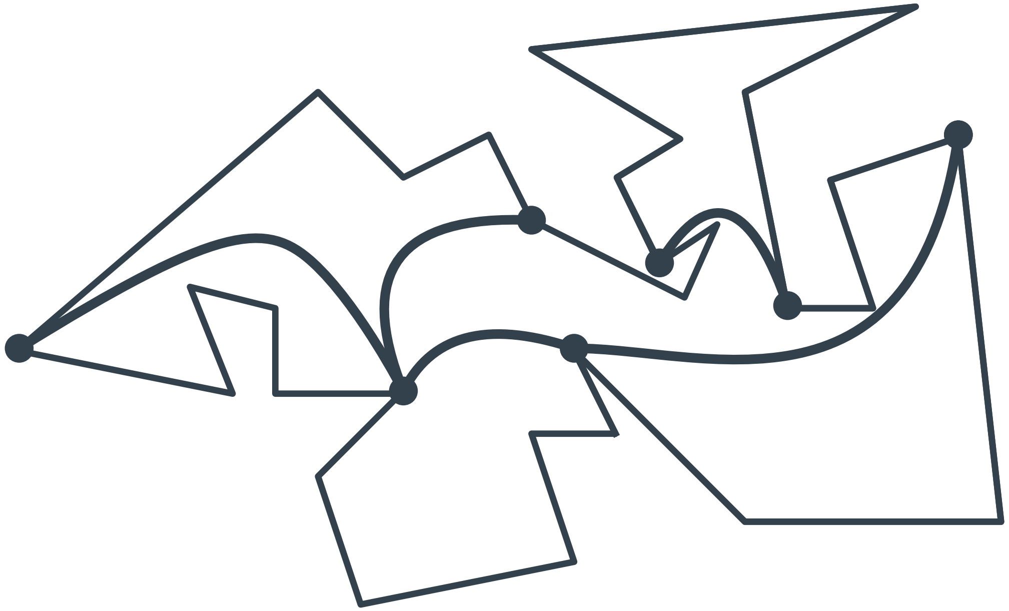
G admits an outerplanar drawing with at most one bend per edge that contains the given drawing of the outer face?

Can be decided in $O(nm)$ time

(n: # vertices, m: # interior edges).

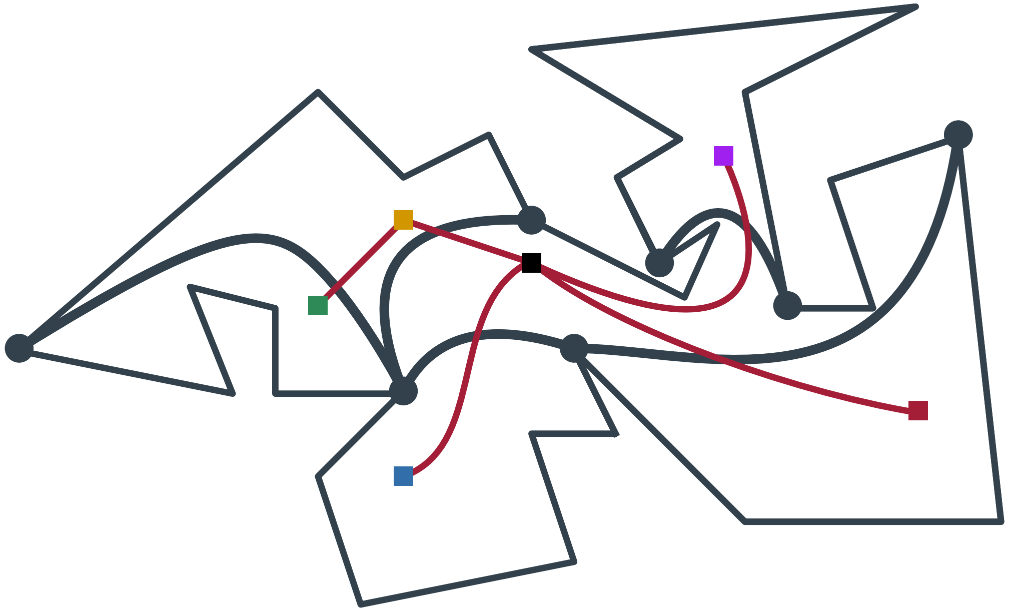
Algorithm overview

- G is a biconnected outerplanar graph ⇒ use dual tree
- Traverse the tree bottom-up, incrementally process edges and refine the polygon P



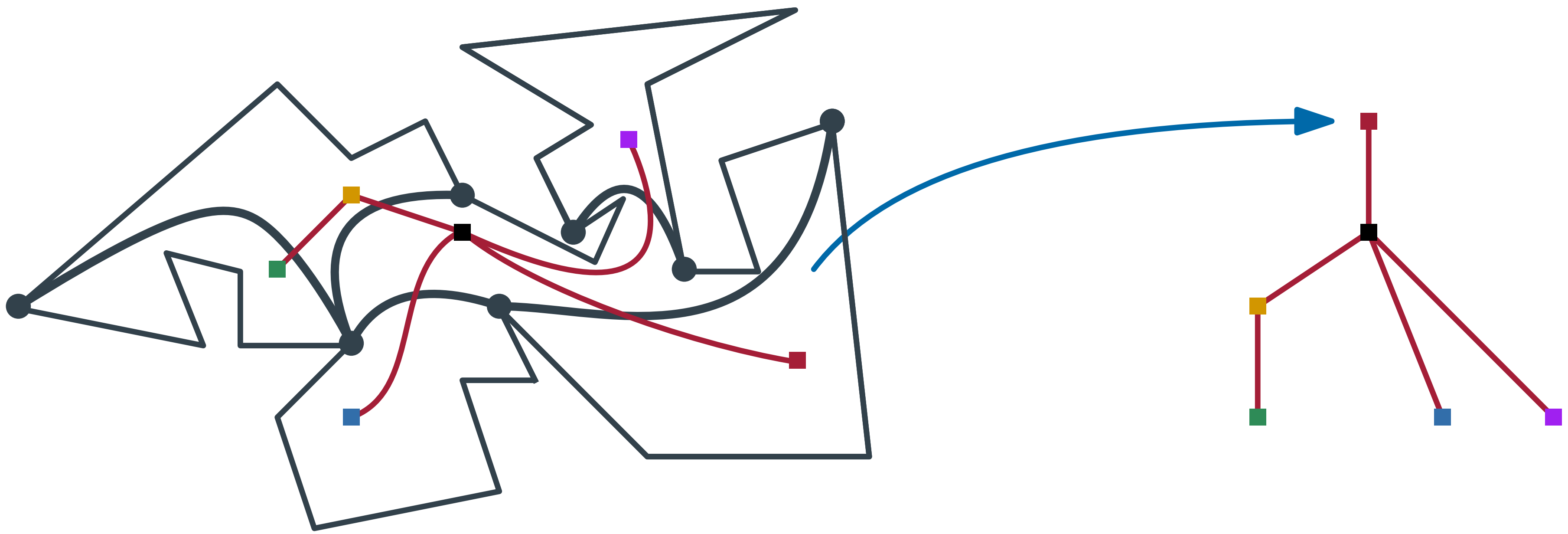
Algorithm overview

- G is a biconnected outerplanar graph $\Rightarrow$ use dual tree
- Traverse the tree bottom-up, incrementally process edges and refine the polygon P



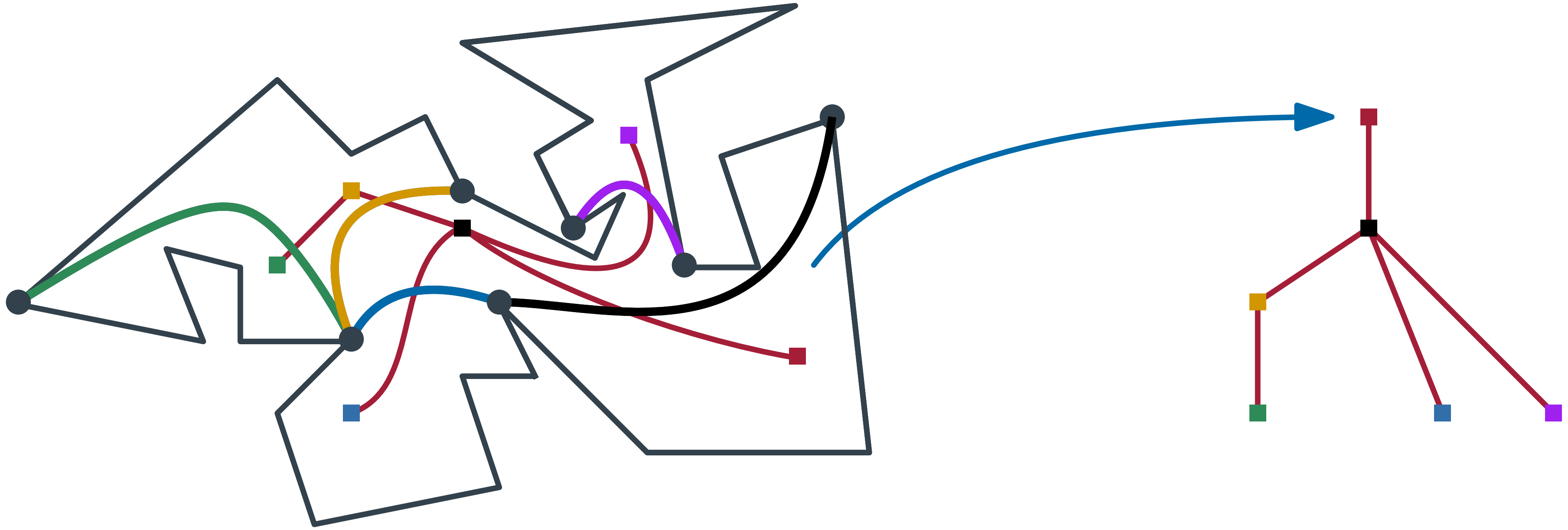
Algorithm overview

- G is a biconnected outerplanar graph $\Rightarrow$ use dual tree
- Traverse the tree bottom-up, incrementally process edges and refine the polygon P



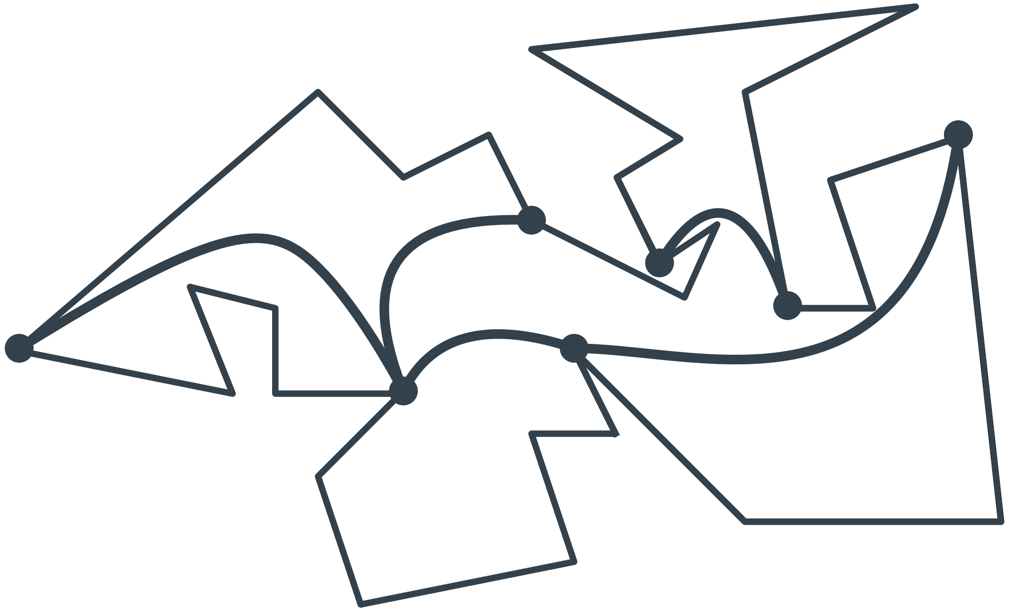
Algorithm overview

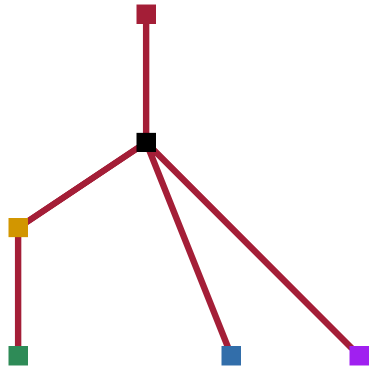
- G is a biconnected outerplanar graph $\Rightarrow$ use dual tree
- Traverse the tree bottom-up, incrementally process edges and refine the polygon P



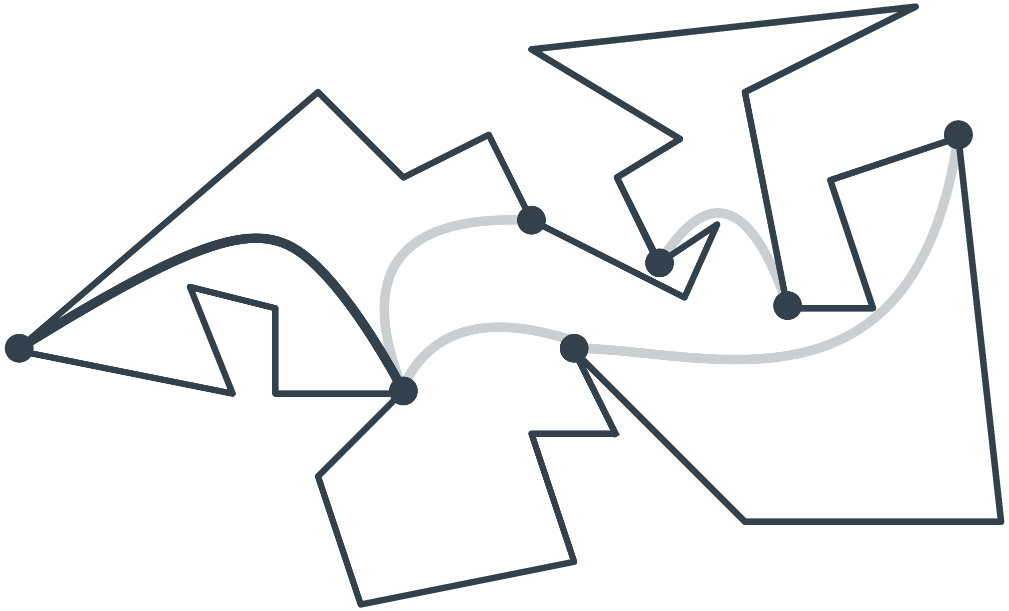
Algorithm overview

- G is a biconnected outerplanar graph ⇒ use dual tree
- Traverse the tree bottom-up, incrementally process edges and refine the polygon P
- Graph fits in polygon ⇔ sub-graph fits in refined polygon
- Place the bend points for the edges in a top-down traversal



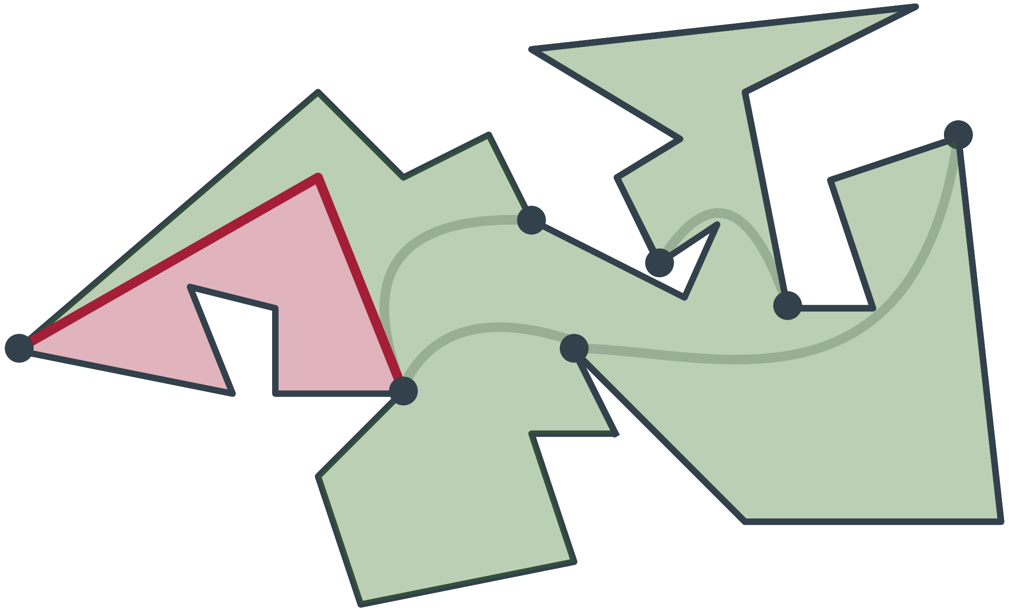


Details: Refine the polygon



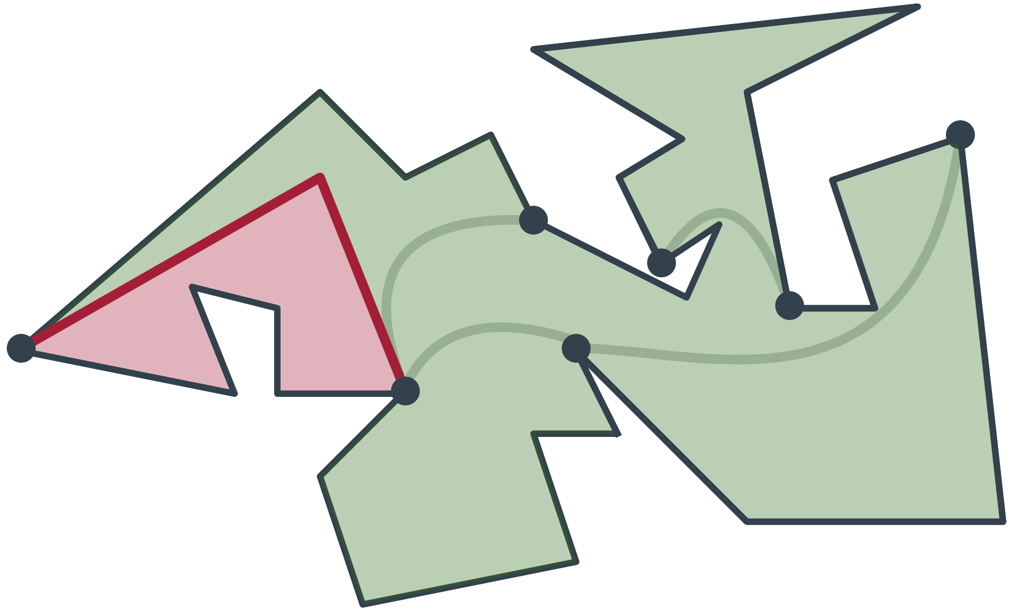
Details: Refine the polygon

- Drawing an edge cuts the polygon into two parts



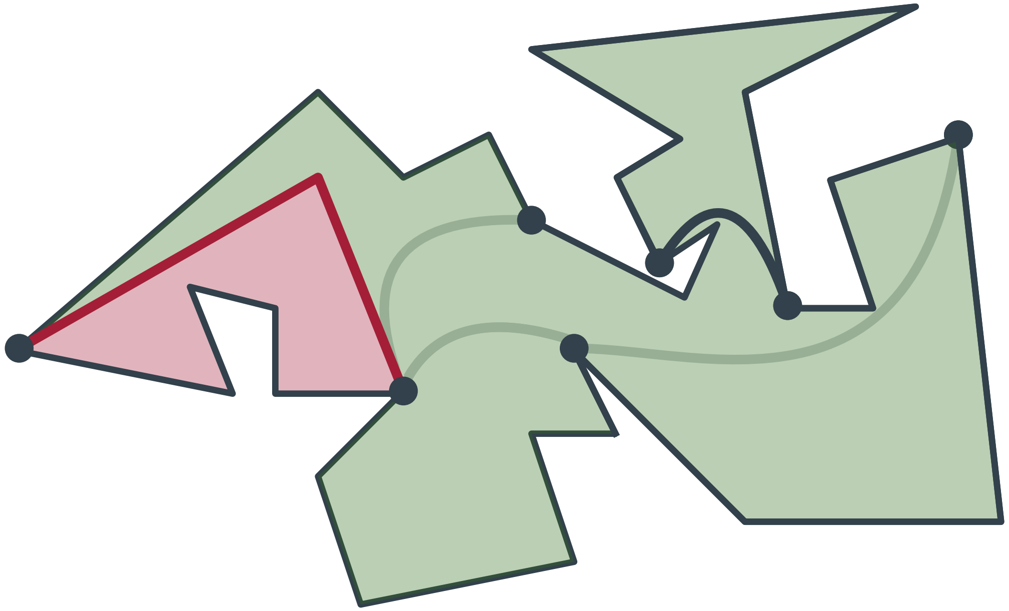
Details: Refine the polygon

- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex



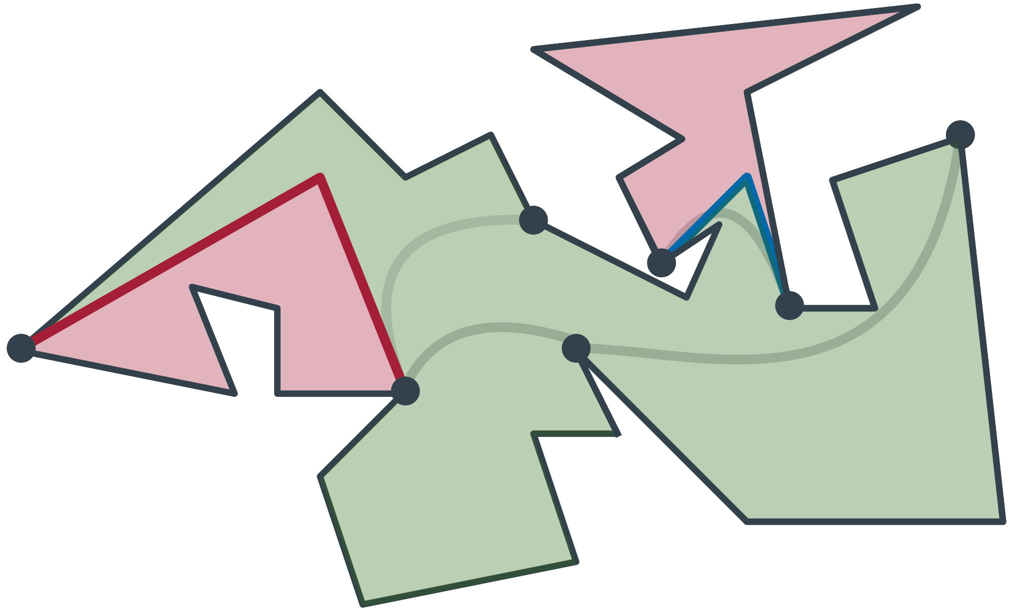
Details: Refine the polygon

- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex



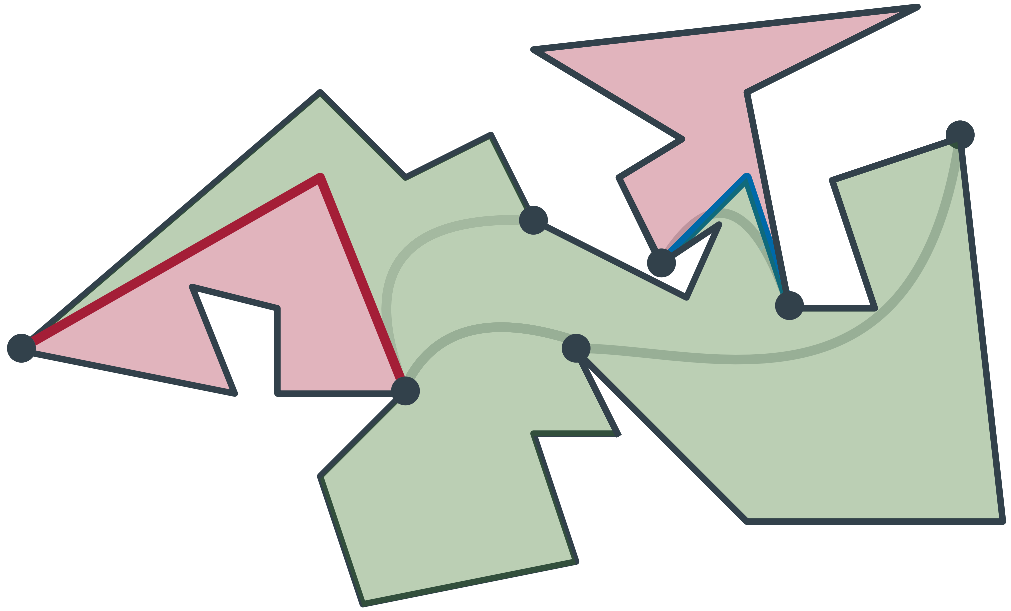
Details: Refine the polygon

- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex



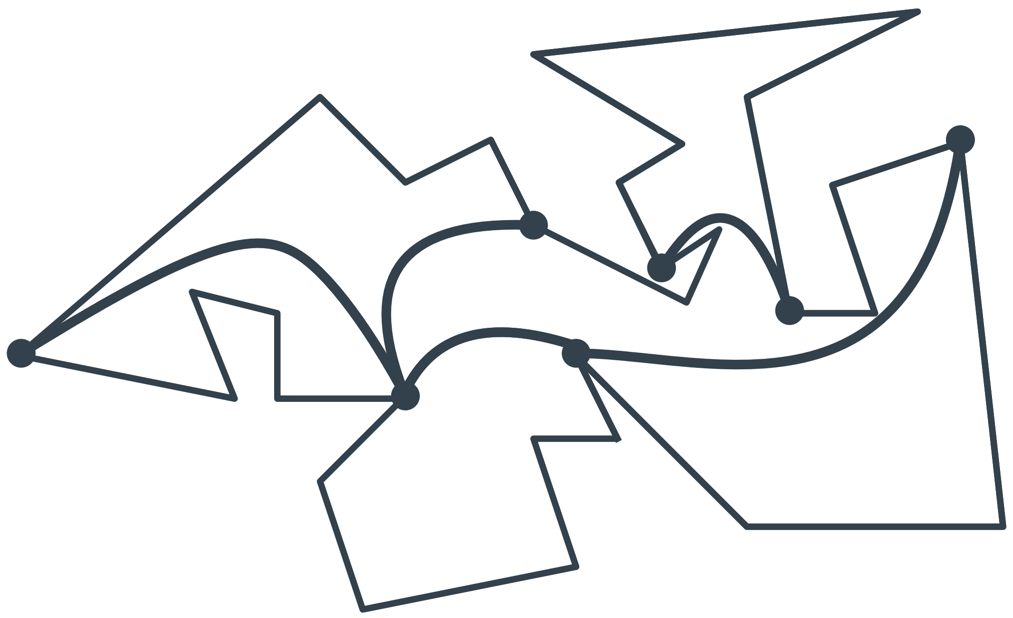
Details: Refine the polygon

- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



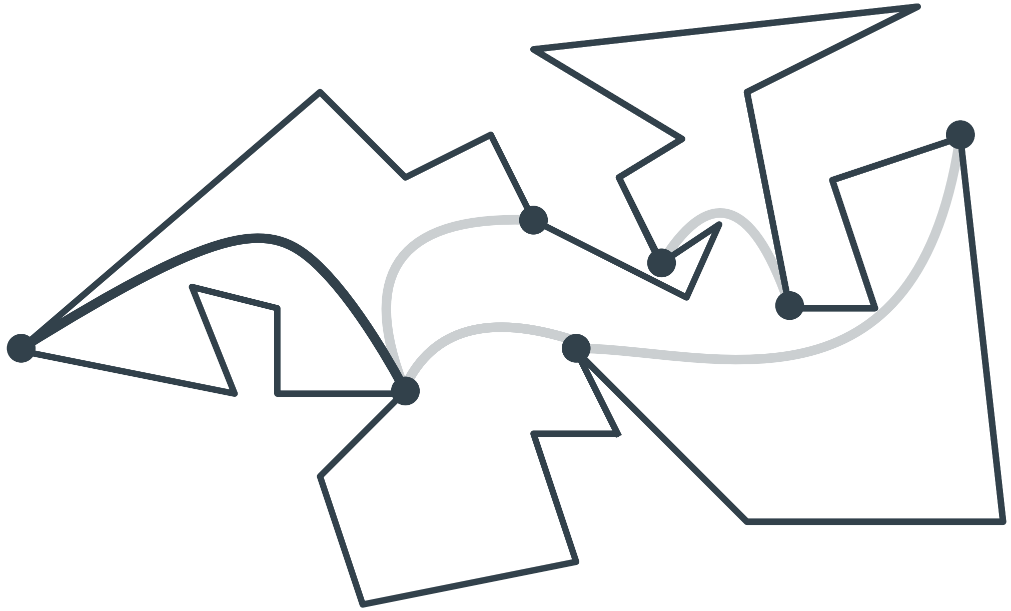
Details: Refine the polygon

- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



Details: Refine the polygon

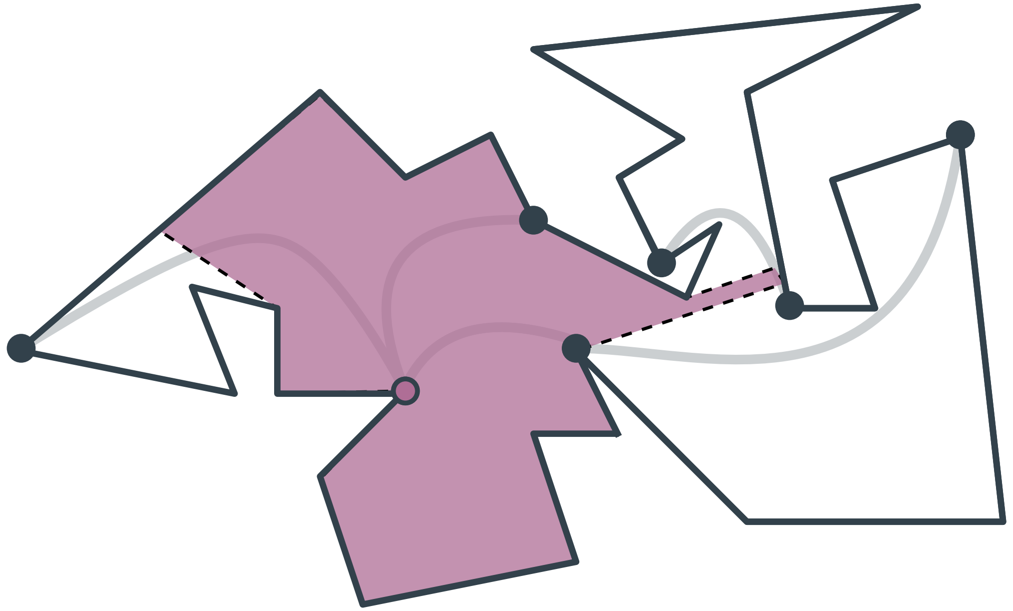
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



reflex

Details: Refine the polygon

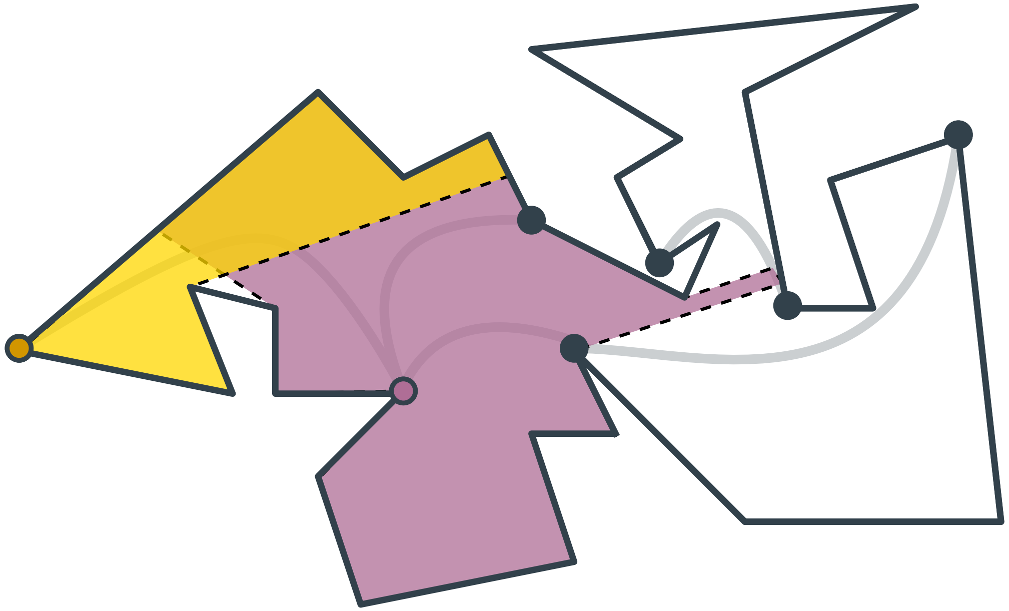
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



reflex

Details: Refine the polygon

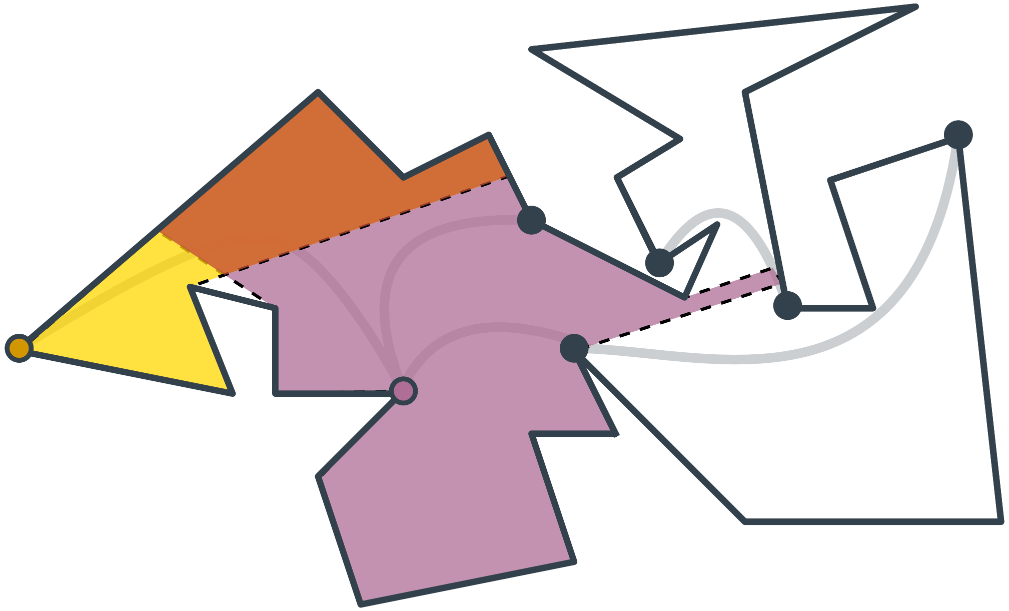
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



reflex

Details: Refine the polygon

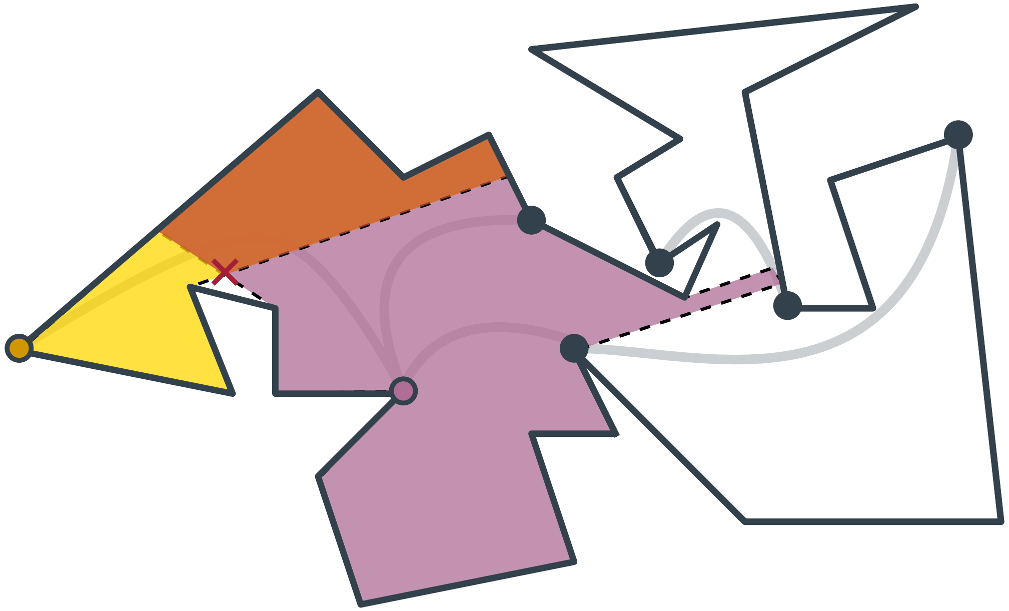
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



reflex

Details: Refine the polygon

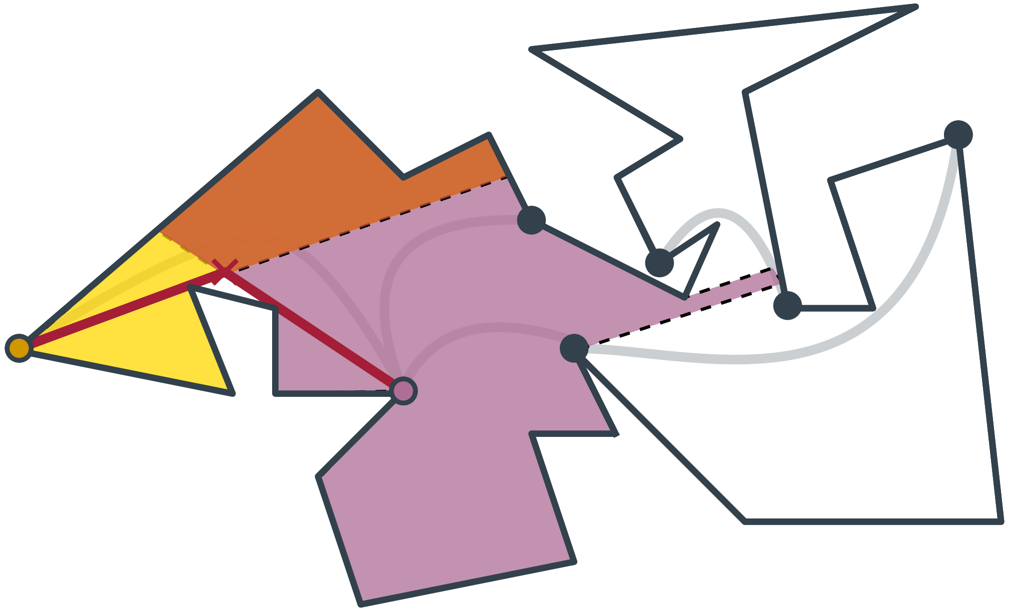
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



reflex
unique "minimal" bend point

Details: Refine the polygon

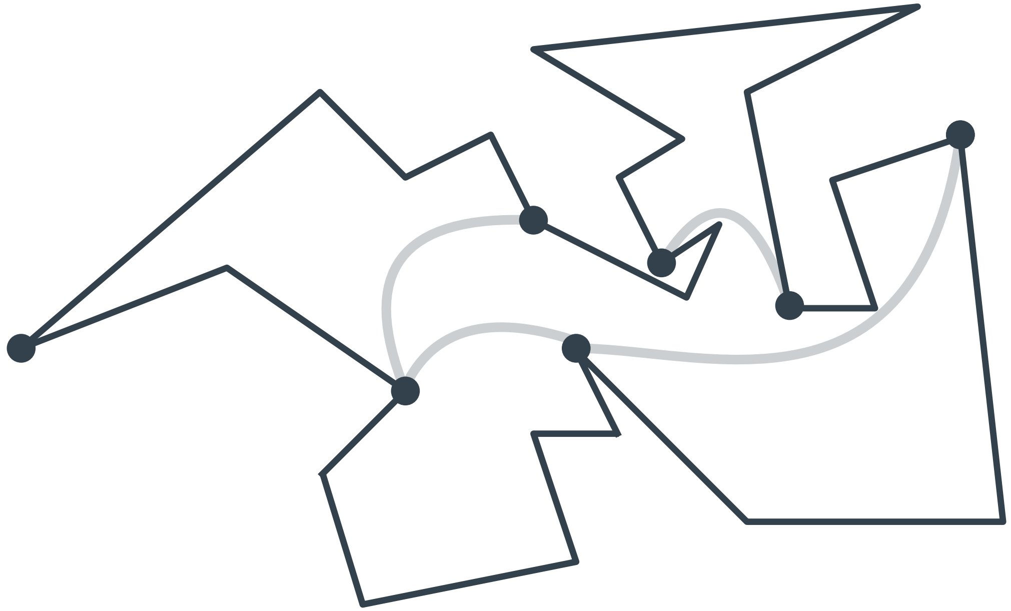
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



reflex
unique "minimal" bend point

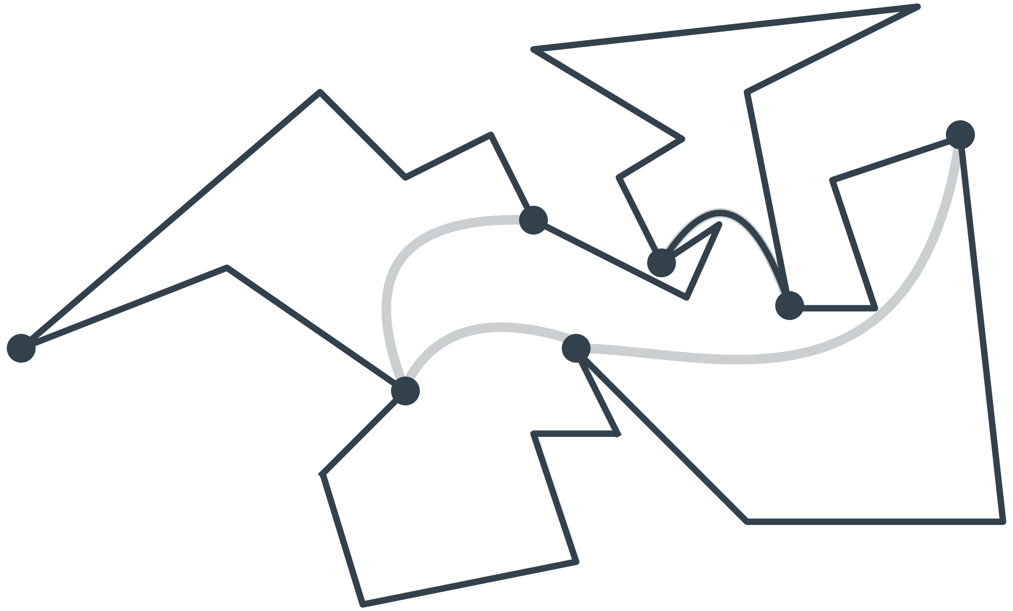
Details: Refine the polygon

- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



Details: Refine the polygon

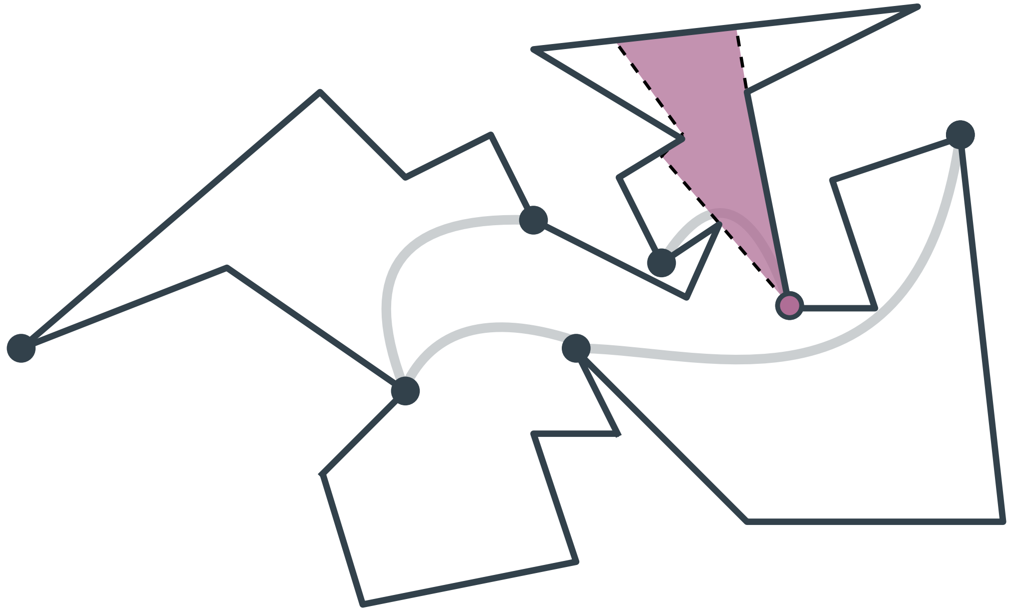
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



convex

Details: Refine the polygon

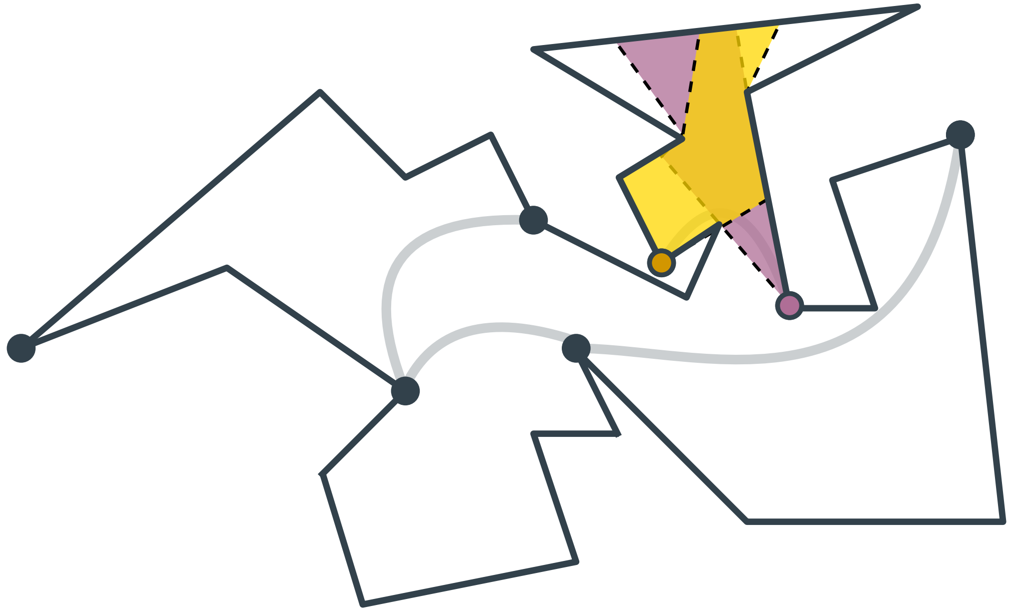
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



convex

Details: Refine the polygon

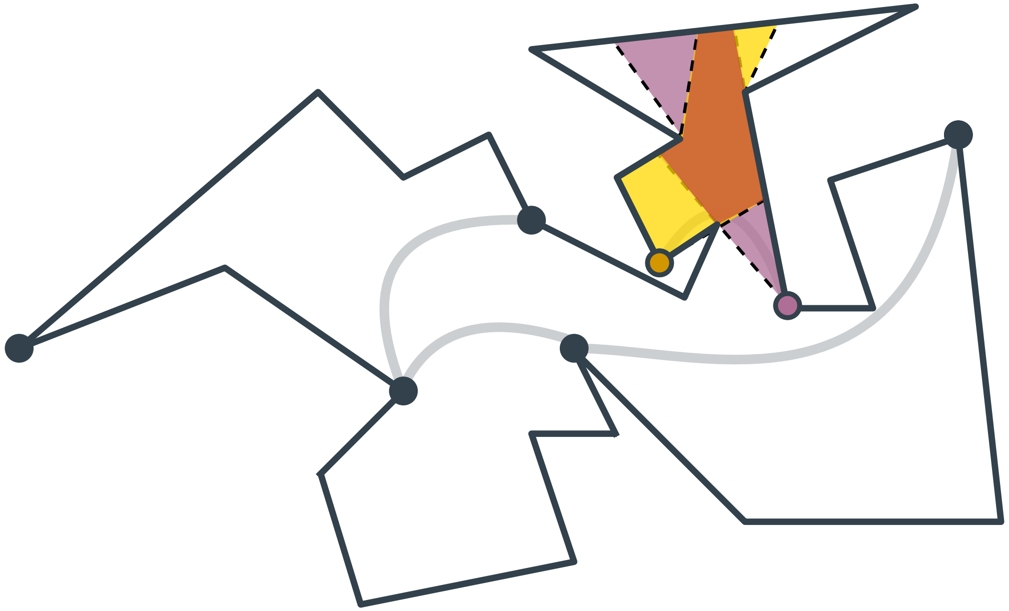
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



convex

Details: Refine the polygon

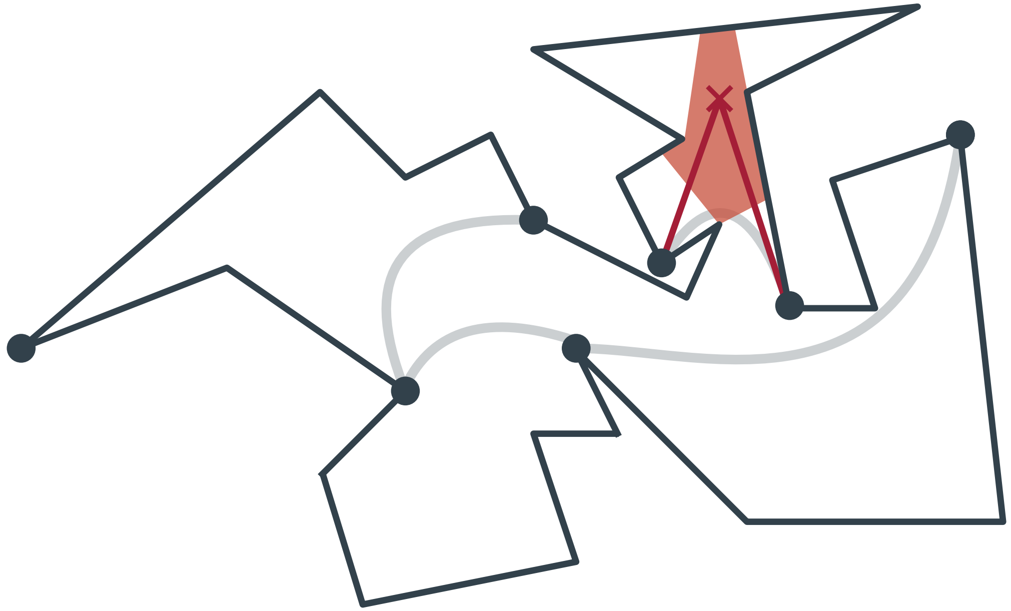
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



convex
wherever the bend point is placed, some points will be cut off for sure

Details: Refine the polygon

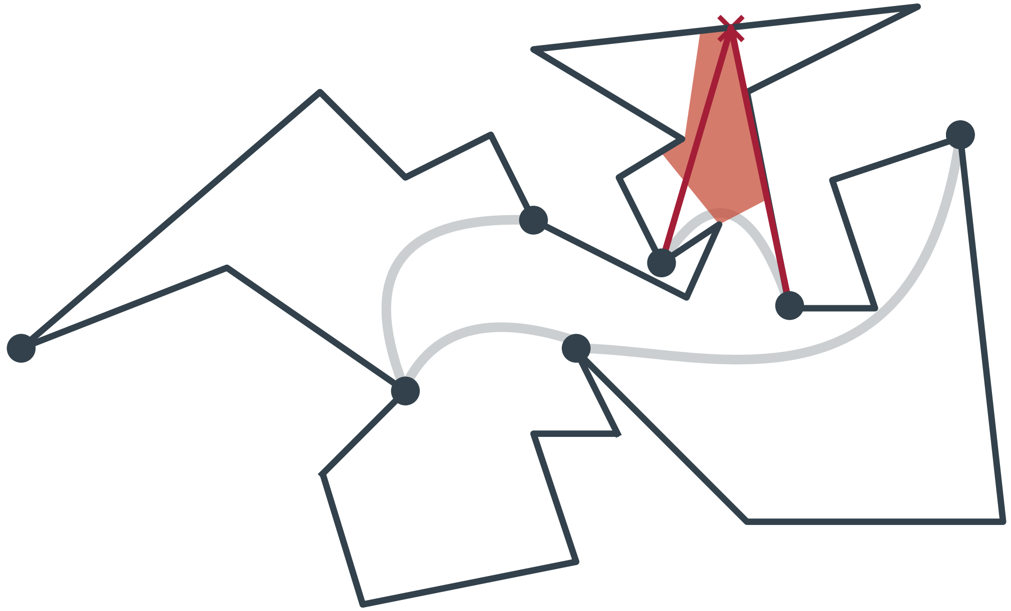
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



convex
wherever the bend point is placed, some points will be cut off for sure

Details: Refine the polygon

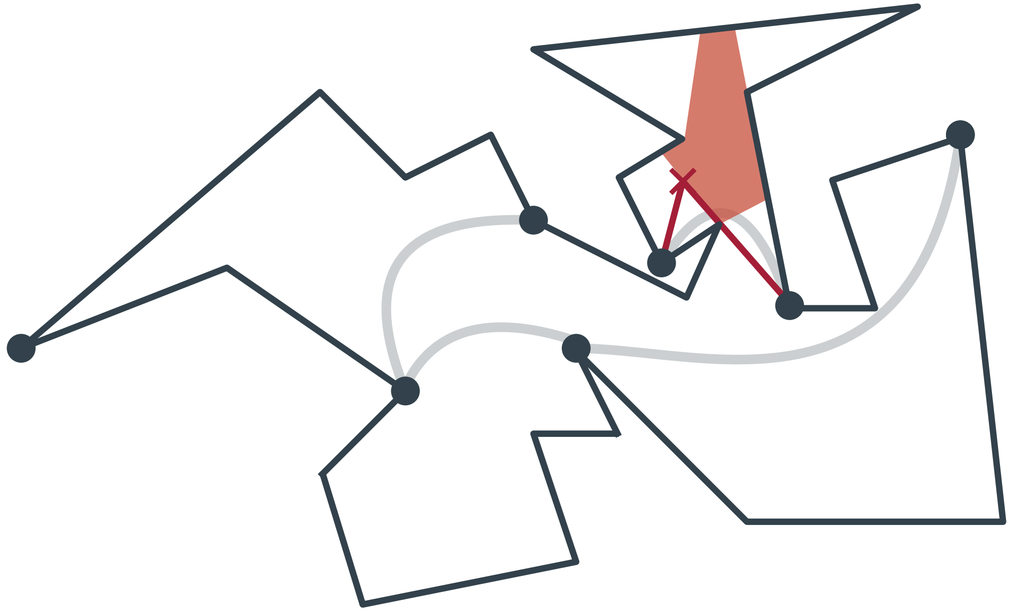
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



convex
wherever the bend point is placed, some points will be cut off for sure

Details: Refine the polygon

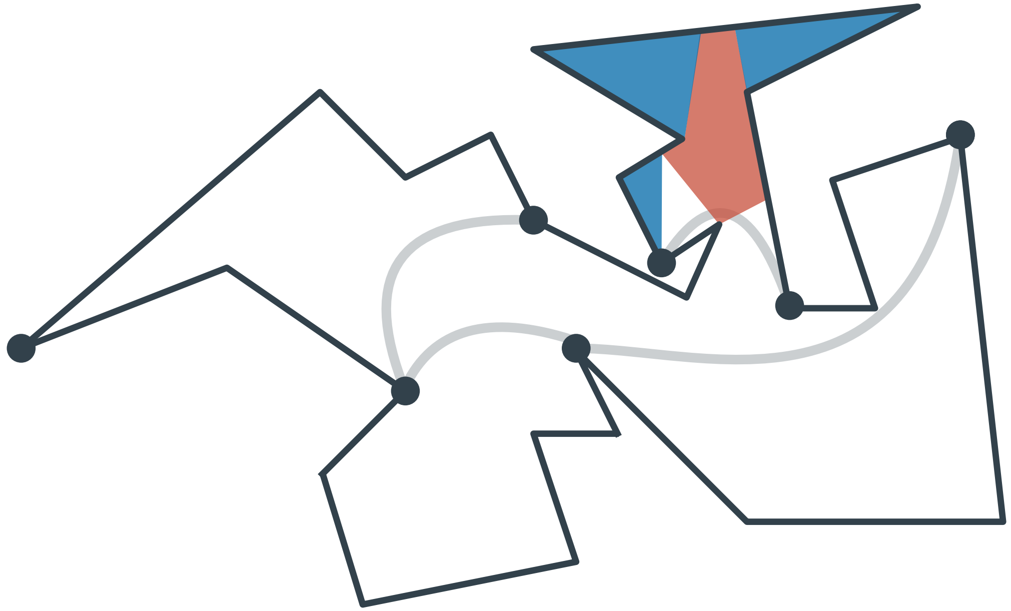
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



convex
wherever the bend point is placed, some points will be cut off for sure

Details: Refine the polygon

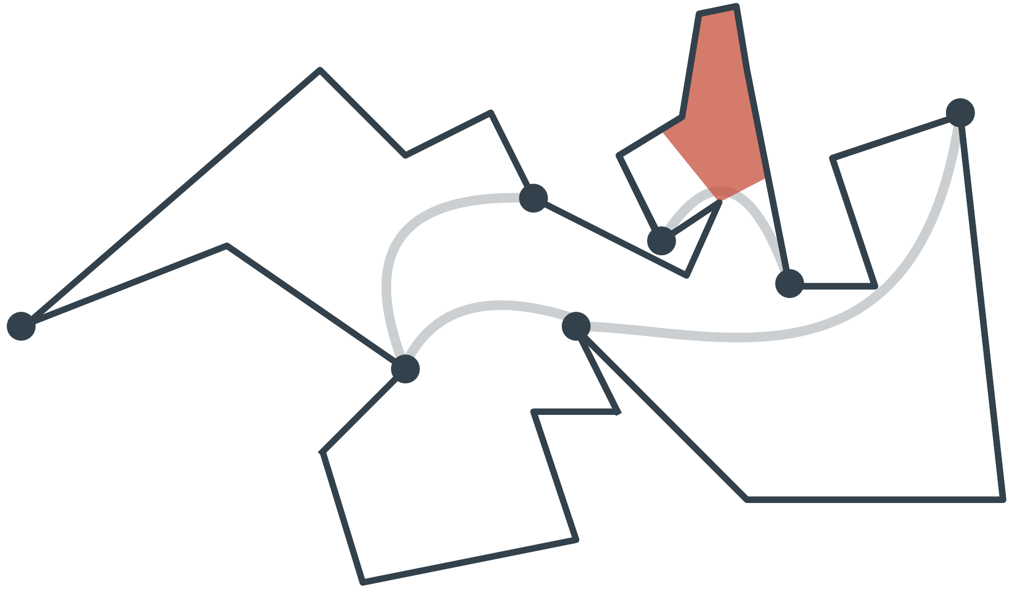
- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible



convex
wherever the bend point is placed, some points will be cut off for sure

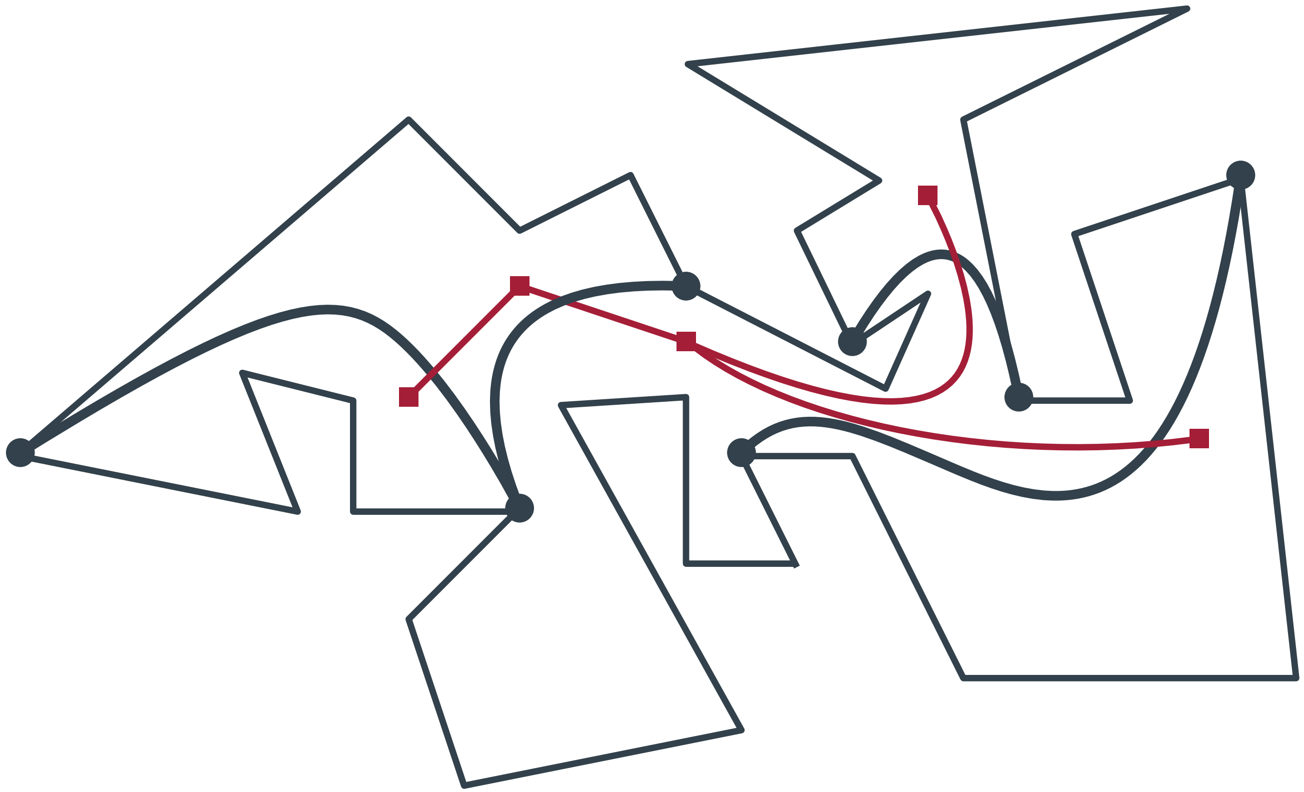
Details: Refine the polygon

- Drawing an edge cuts the polygon into two parts
- Classify the edges: reflex or convex
- Refine the polygon in the least restrictive way: cut off as little as possible

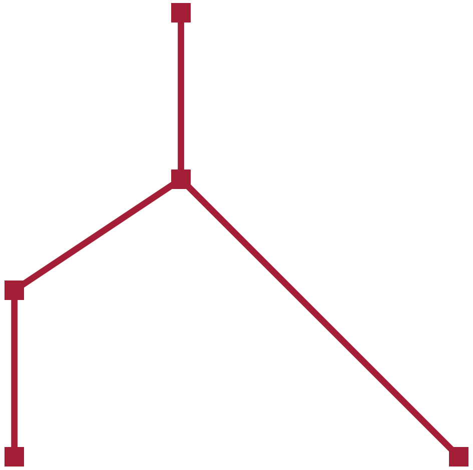


convex
wherever the bend point is placed, some points will be cut off for sure

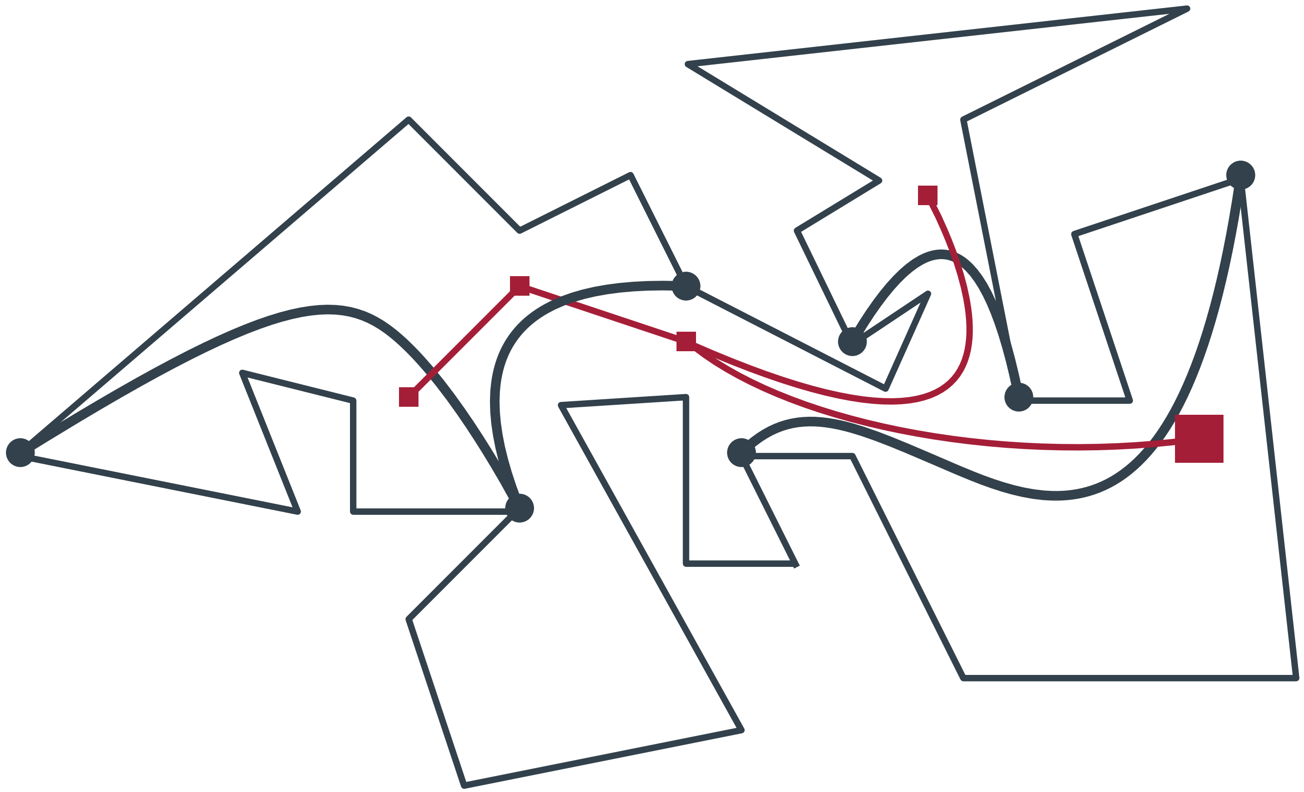
Example



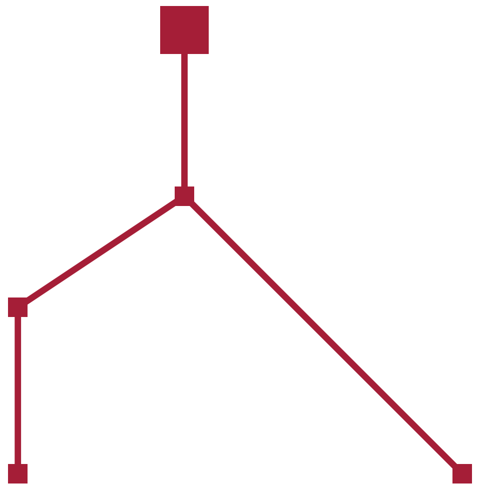
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



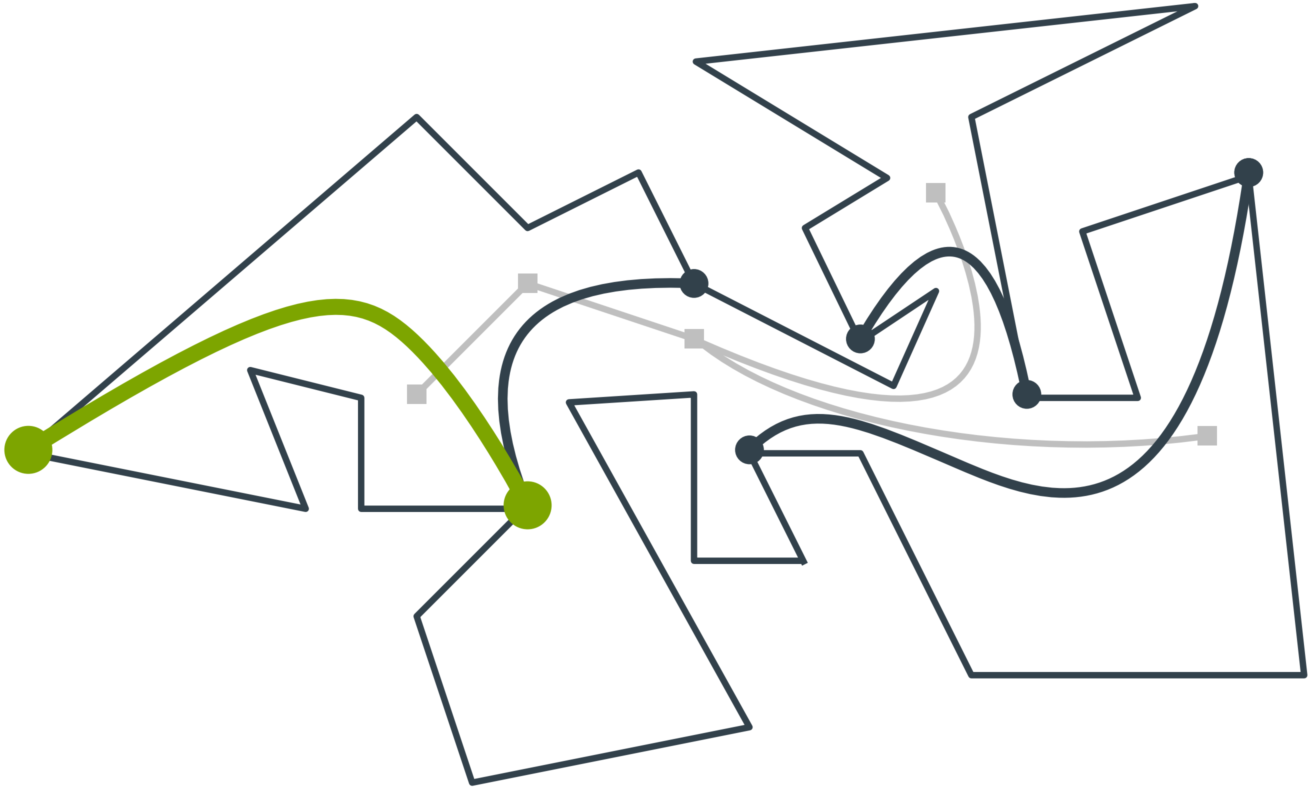
Example



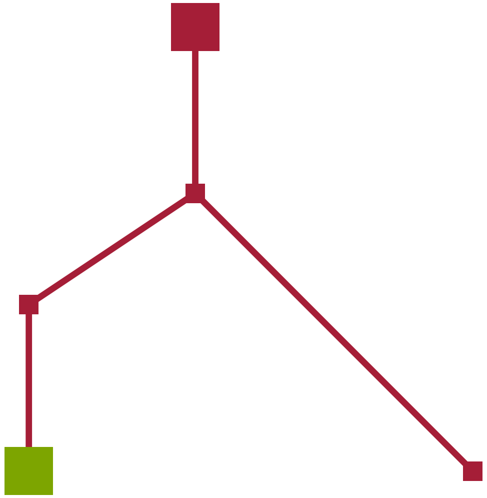
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



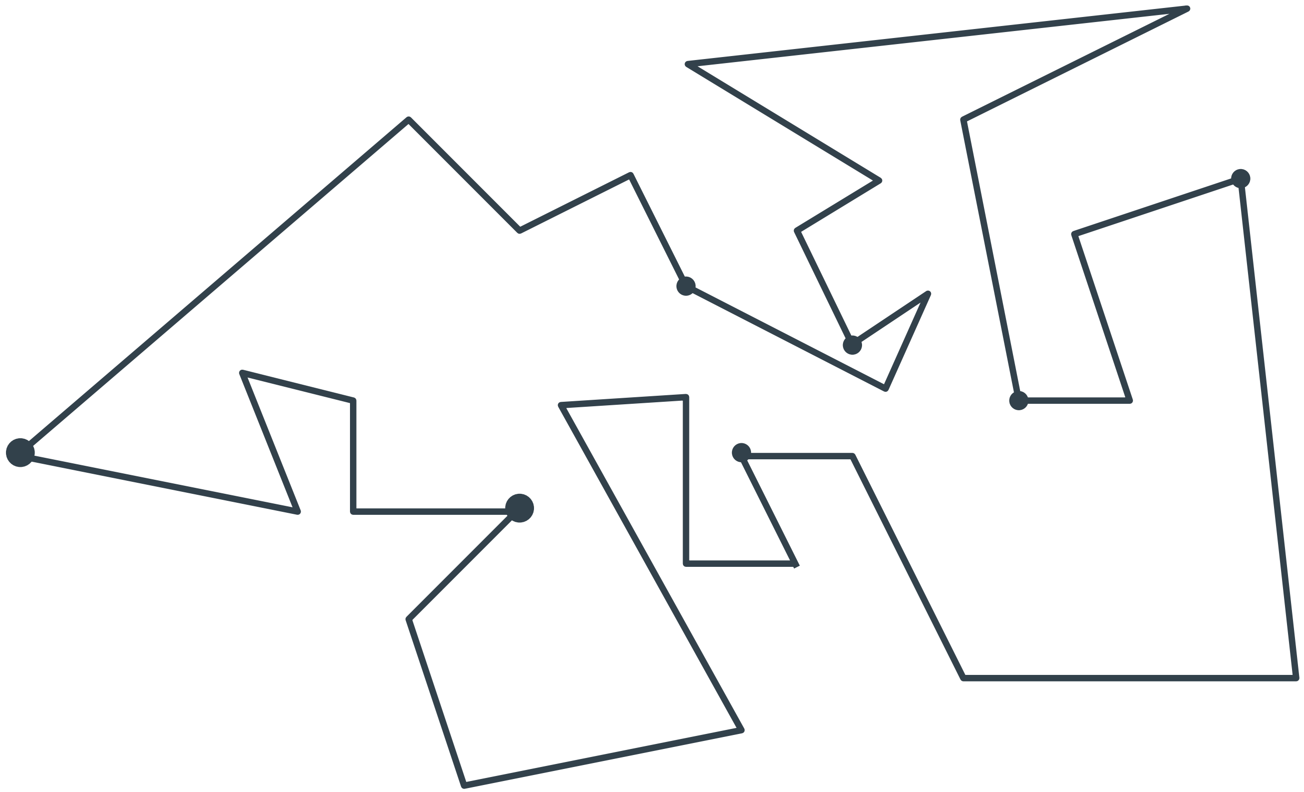
Example



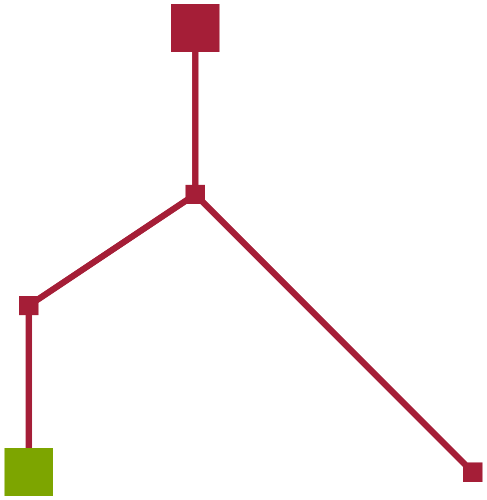
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



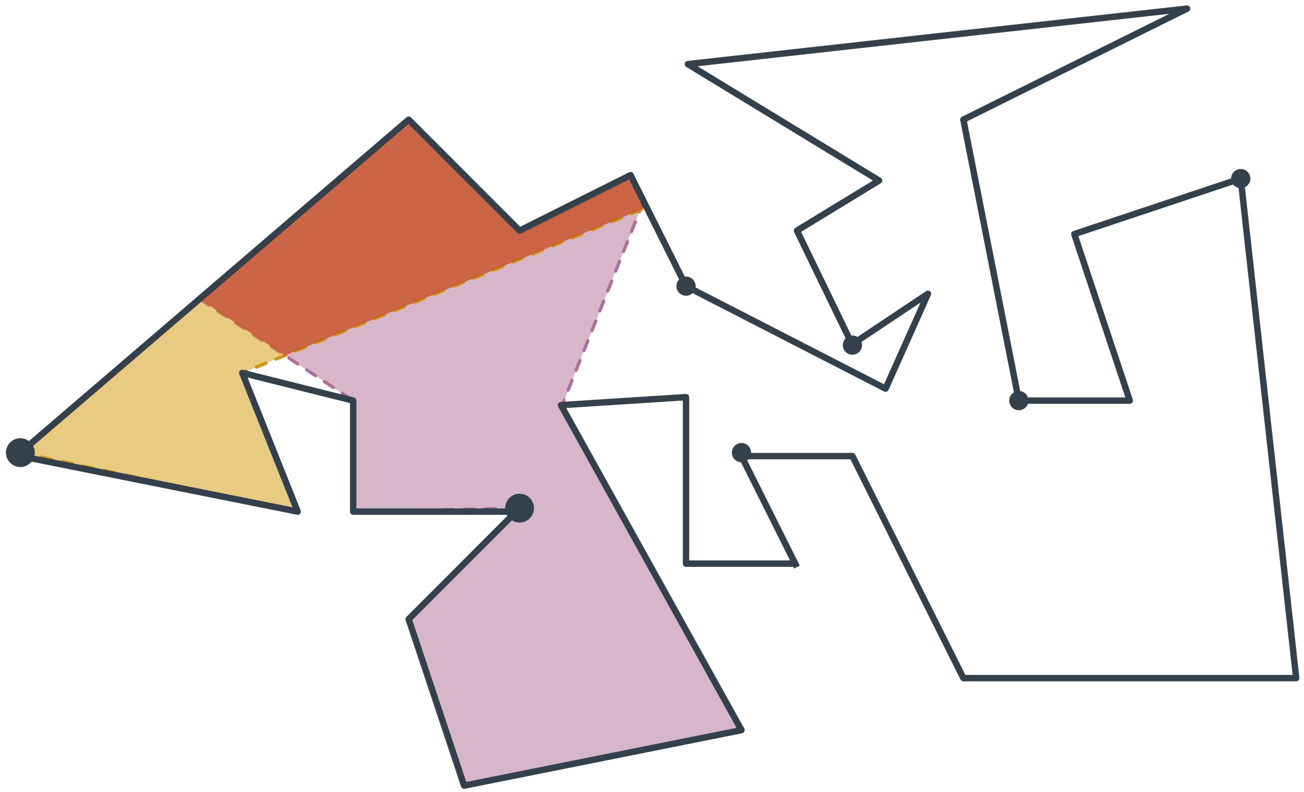
Example



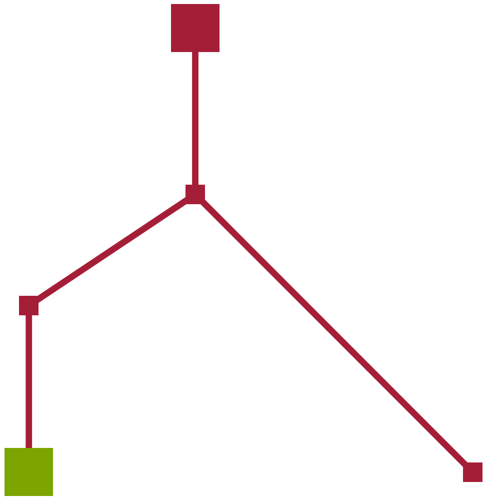
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



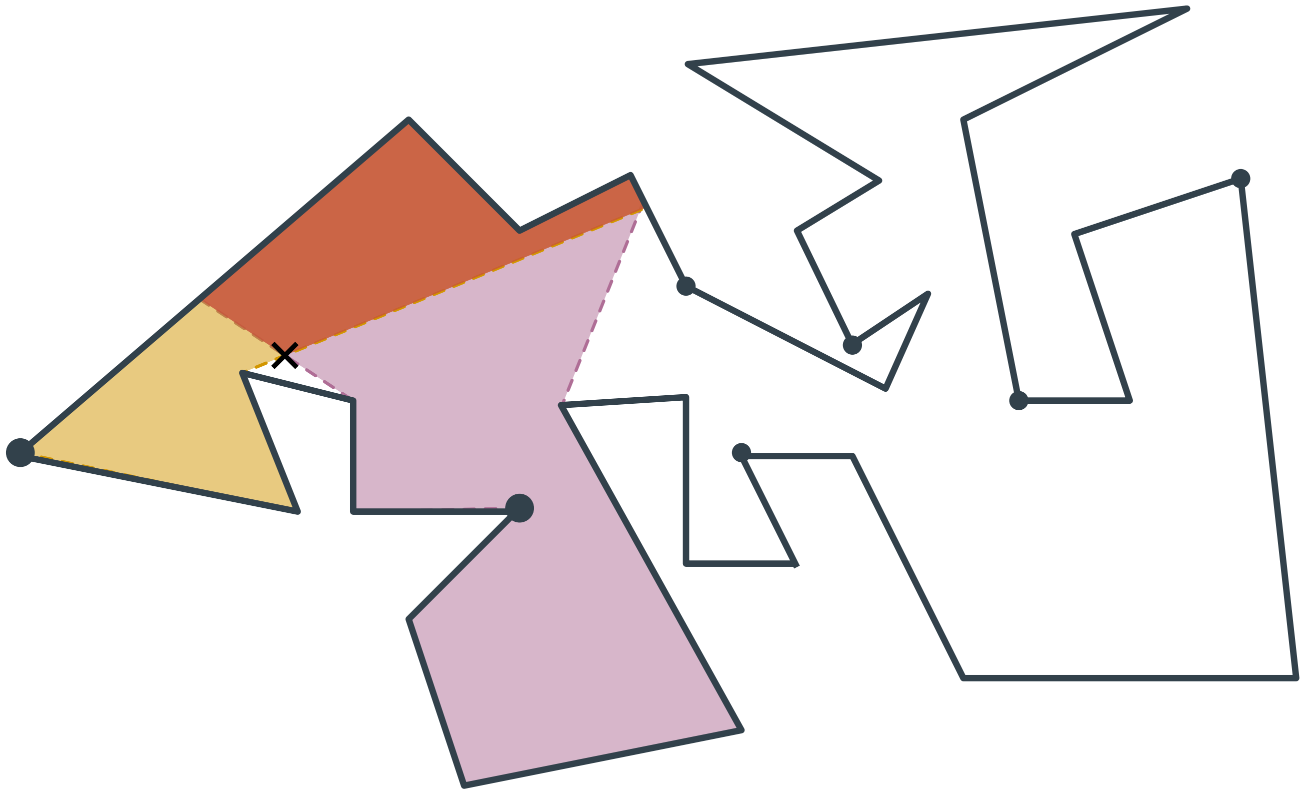
Example

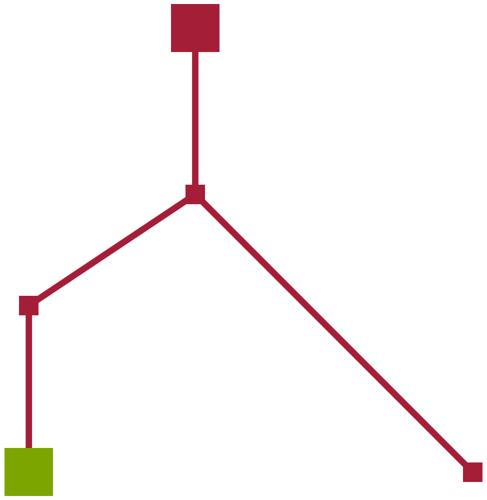


- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



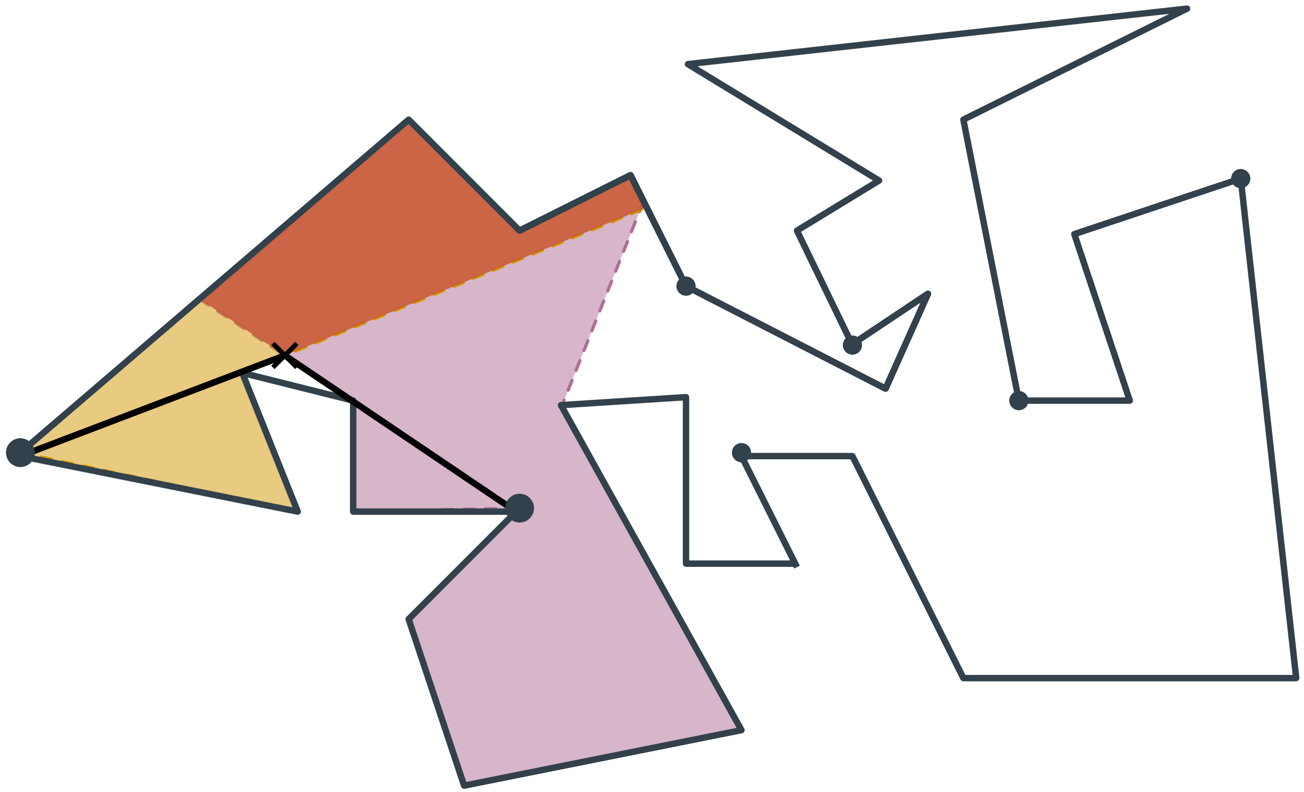
Example



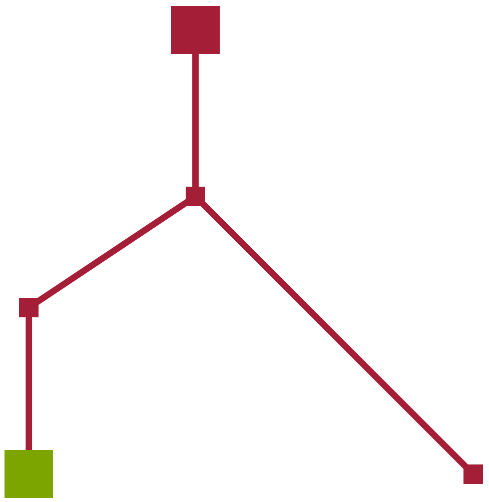


- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons

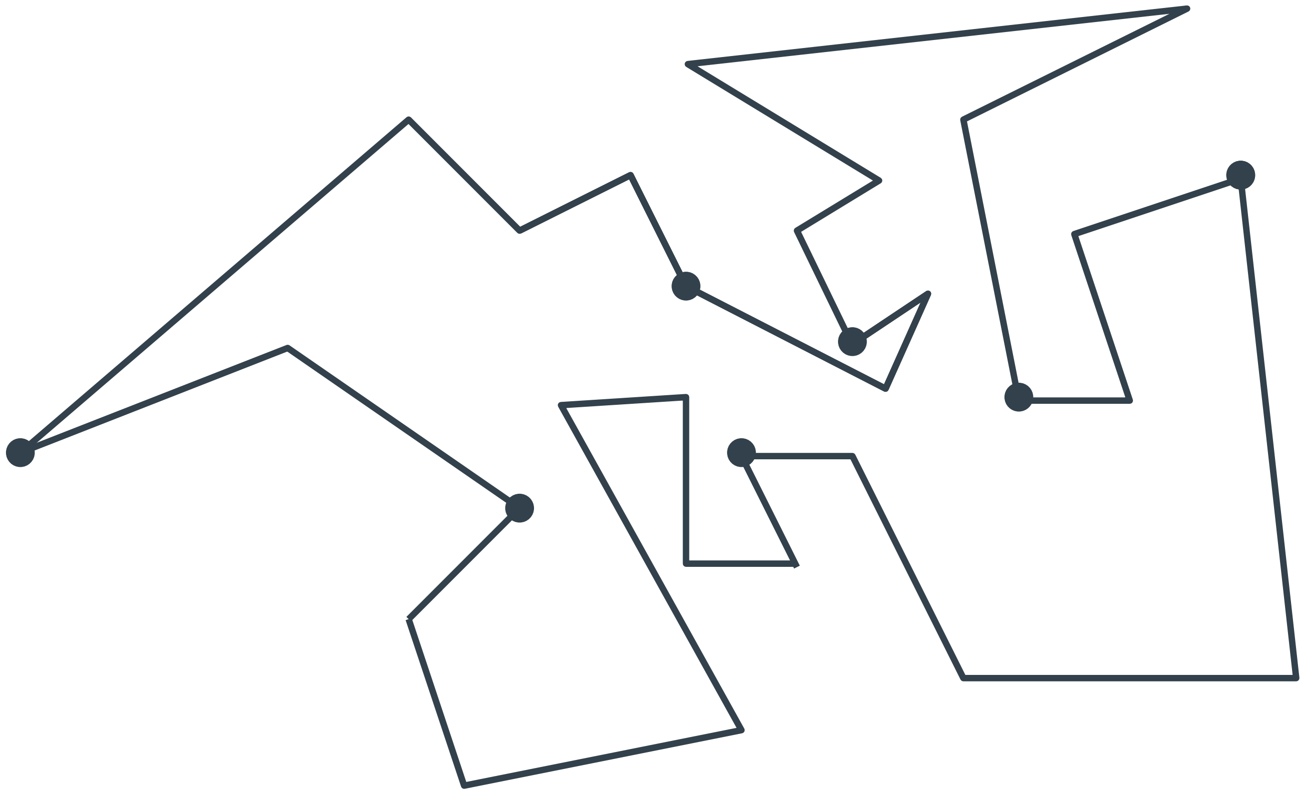
Example

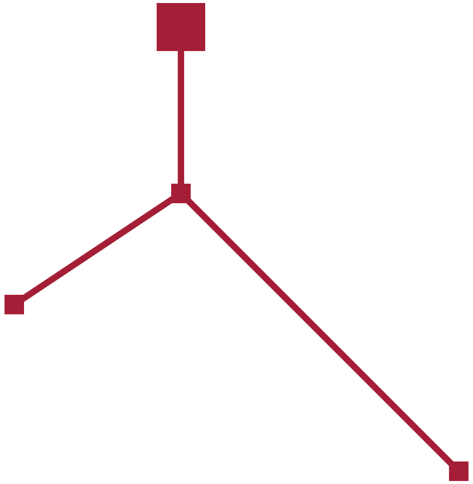


- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



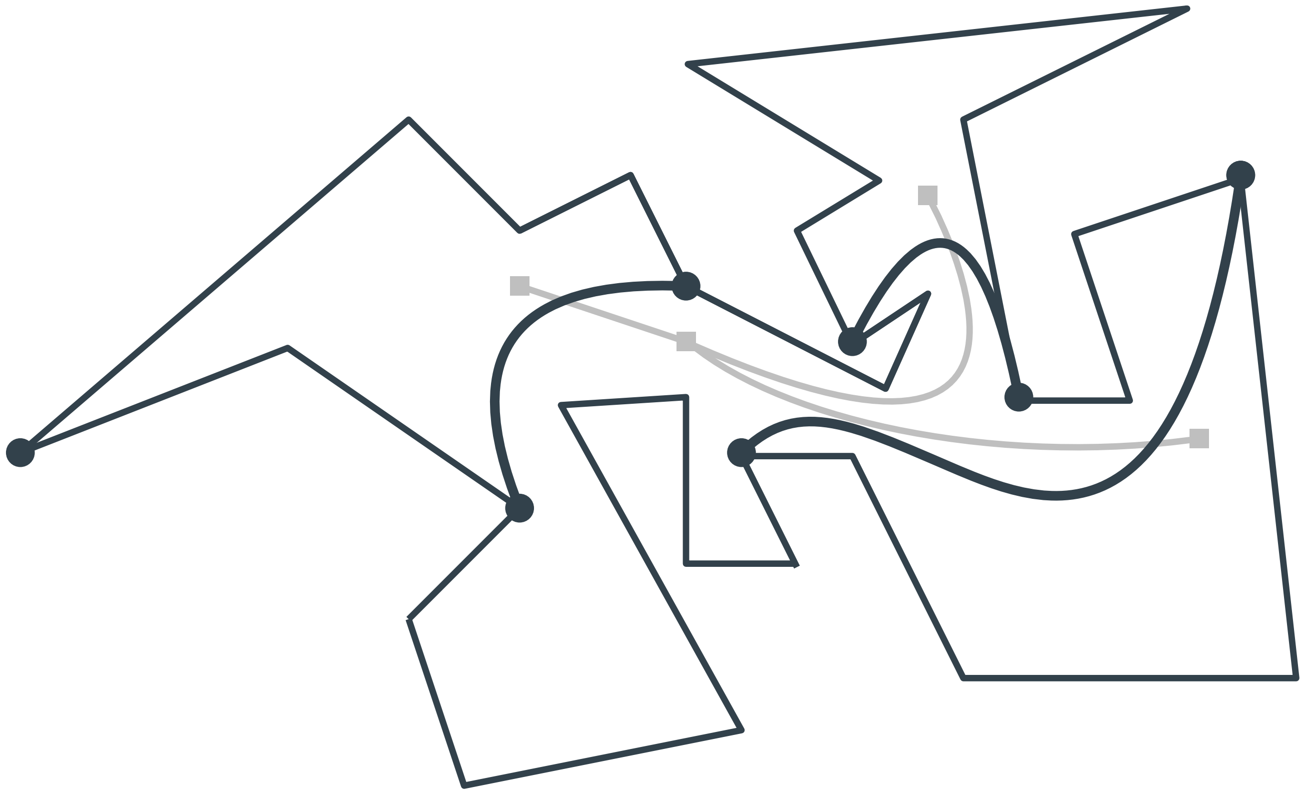
Example



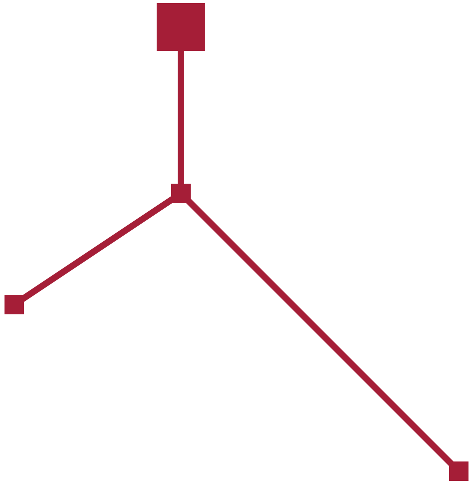


- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons

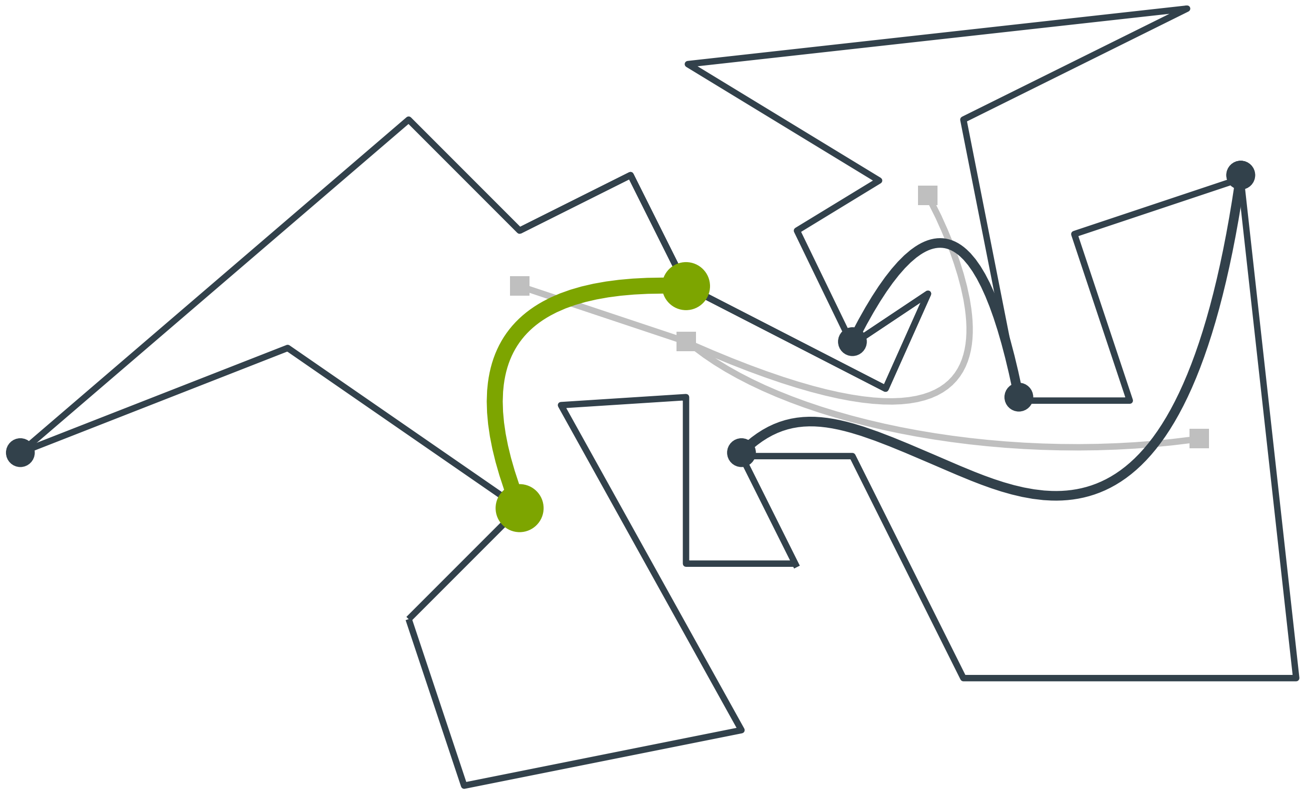
Example



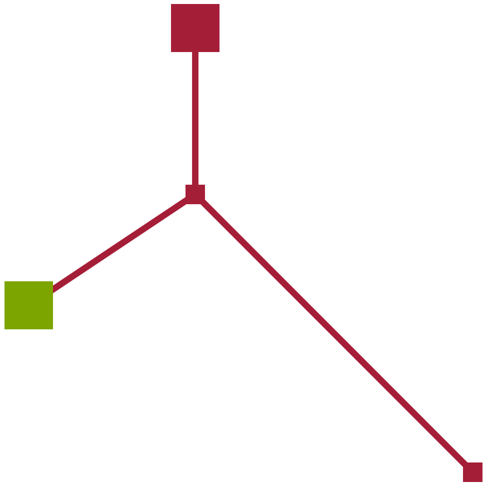
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



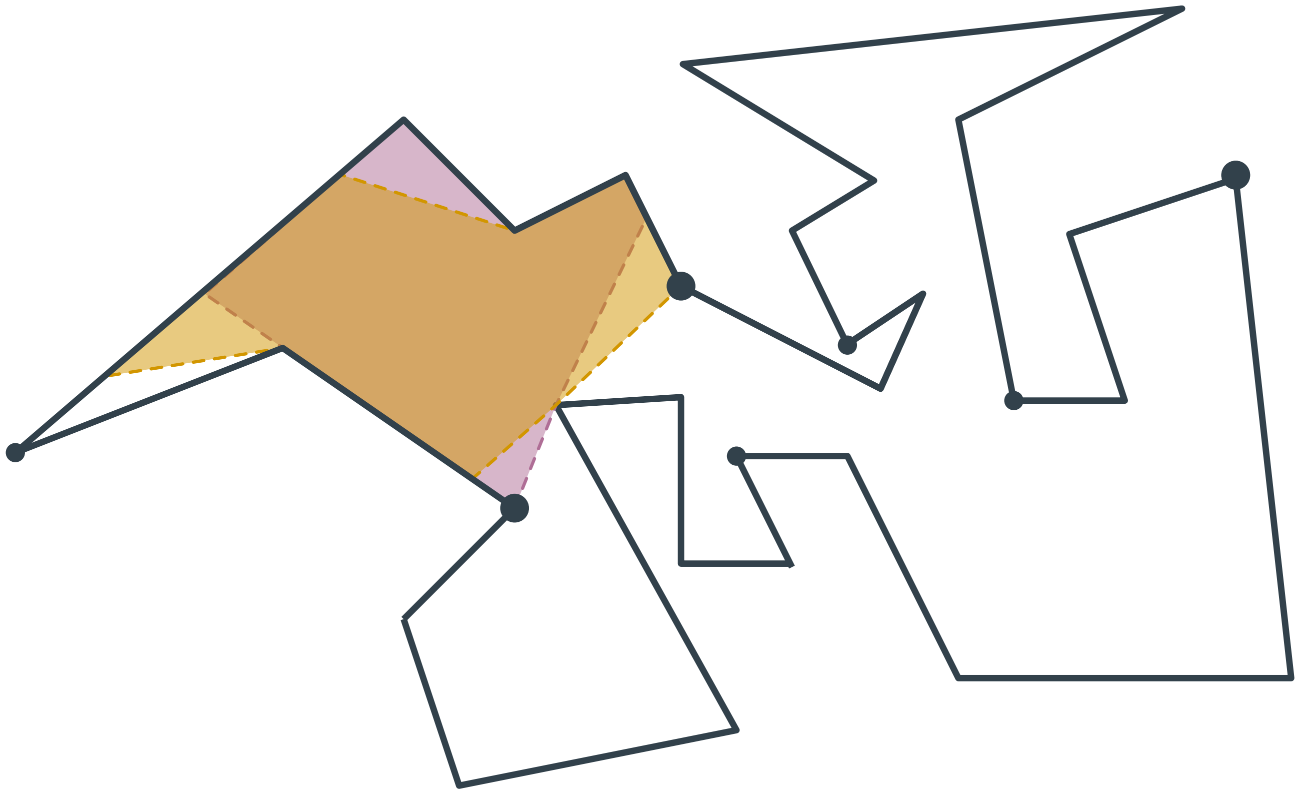
Example



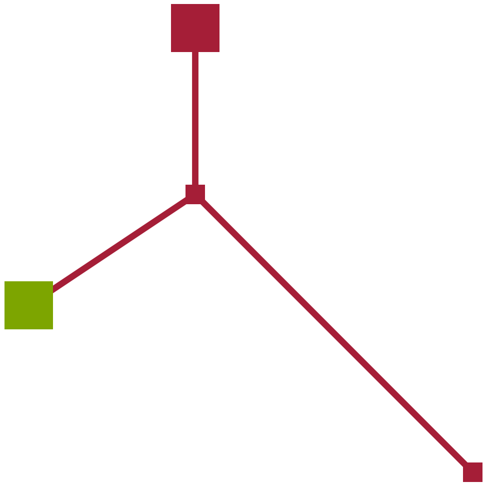
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



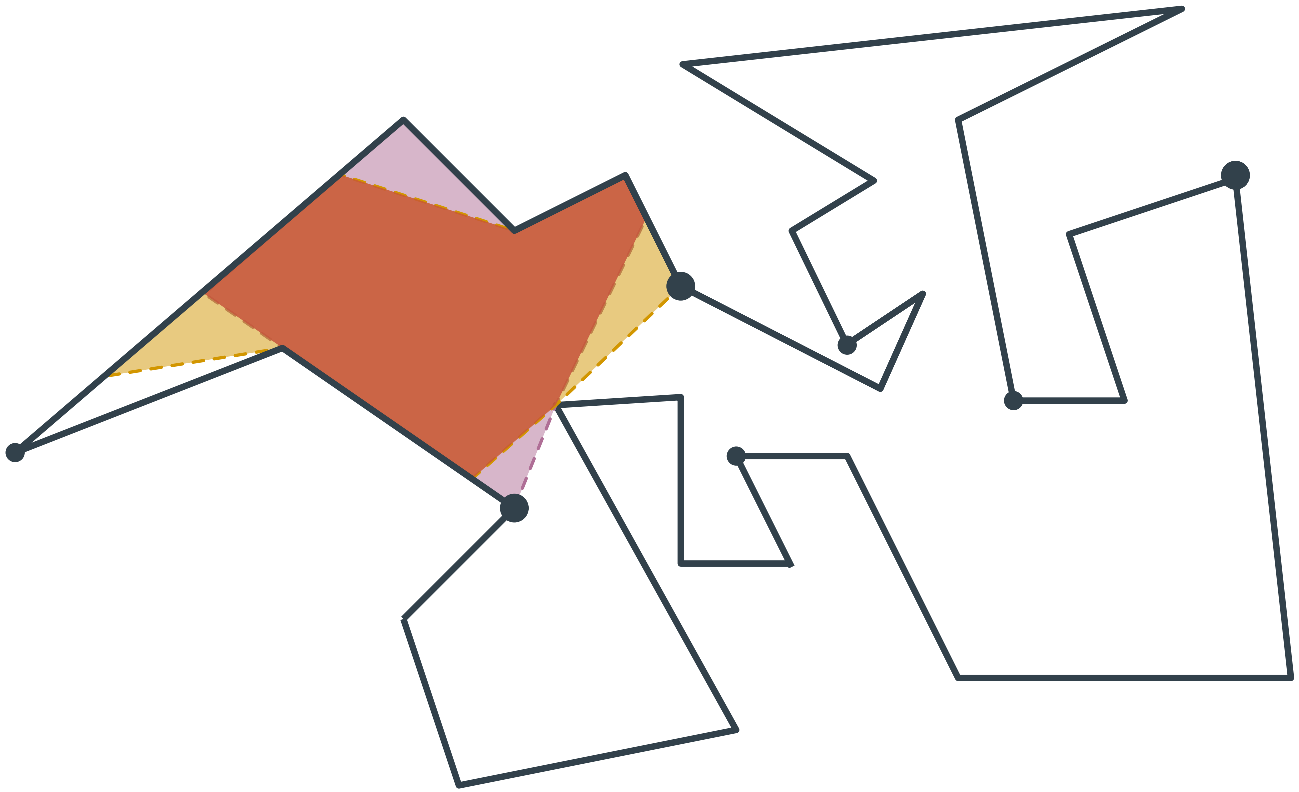
Example

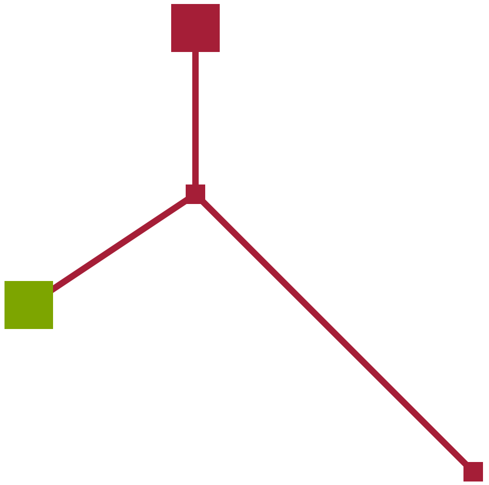


- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



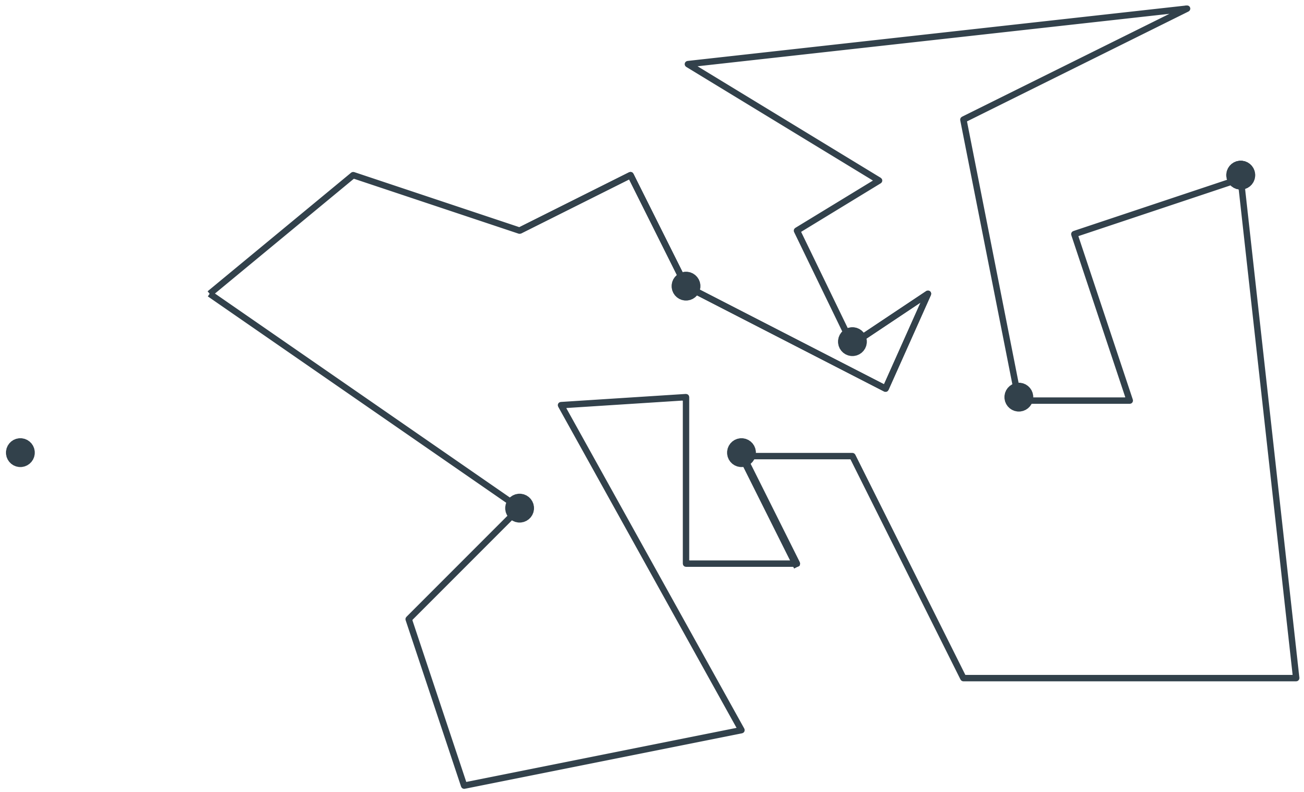
Example



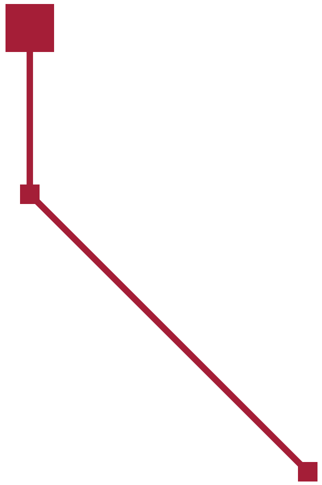


- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons

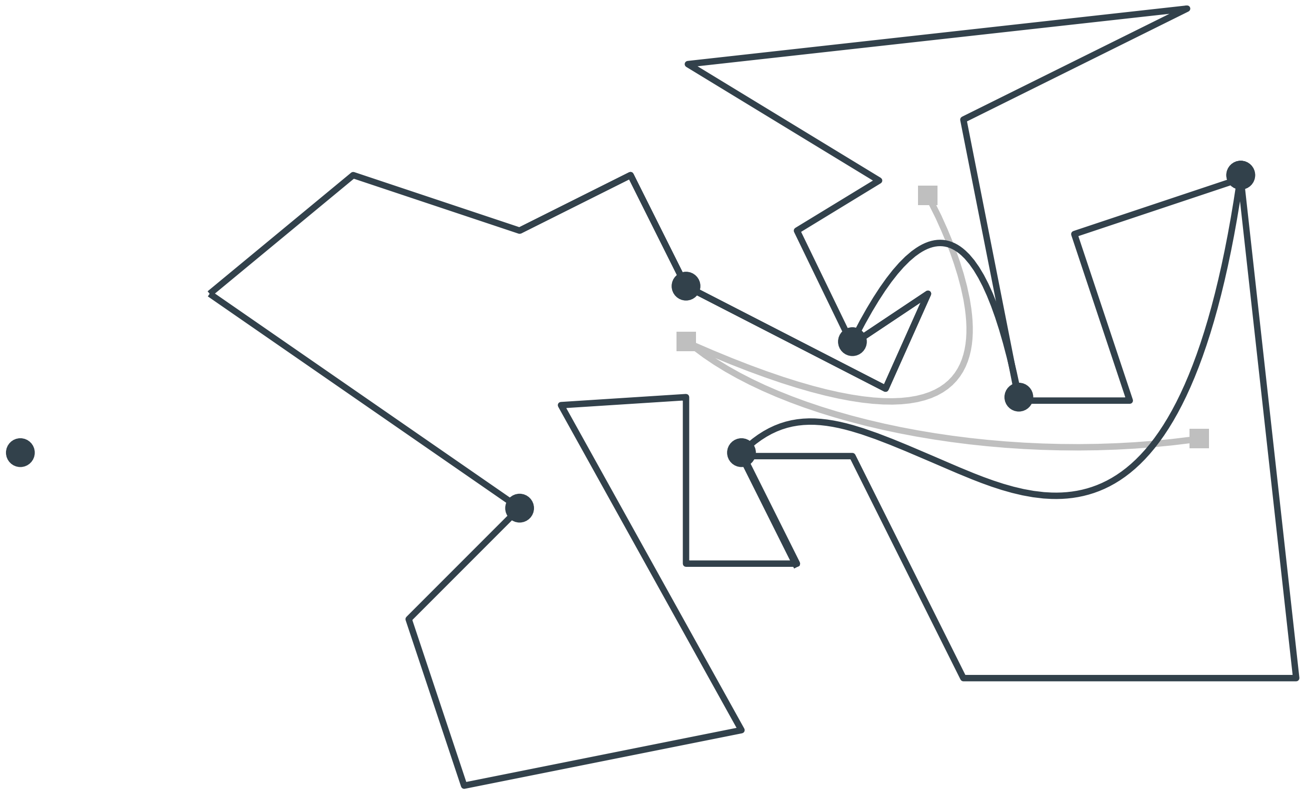
Example



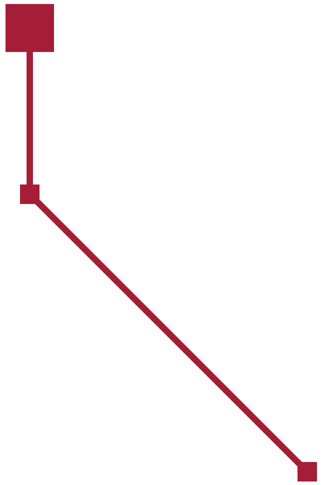
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



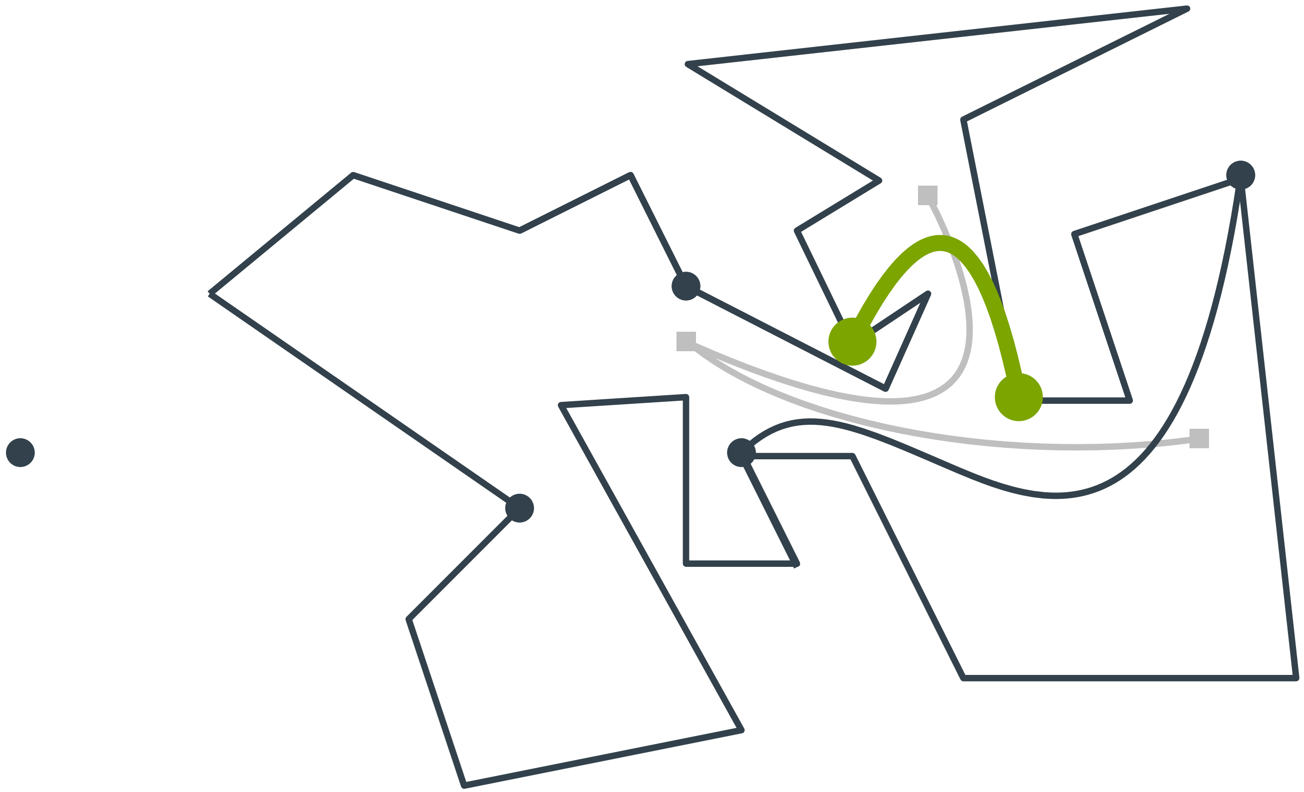
Example



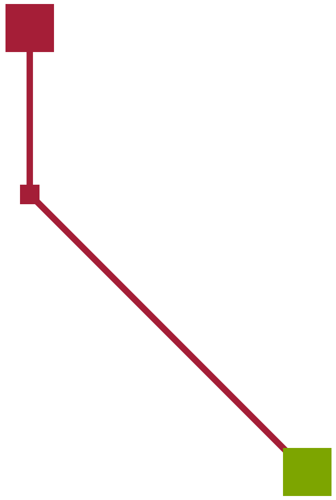
- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



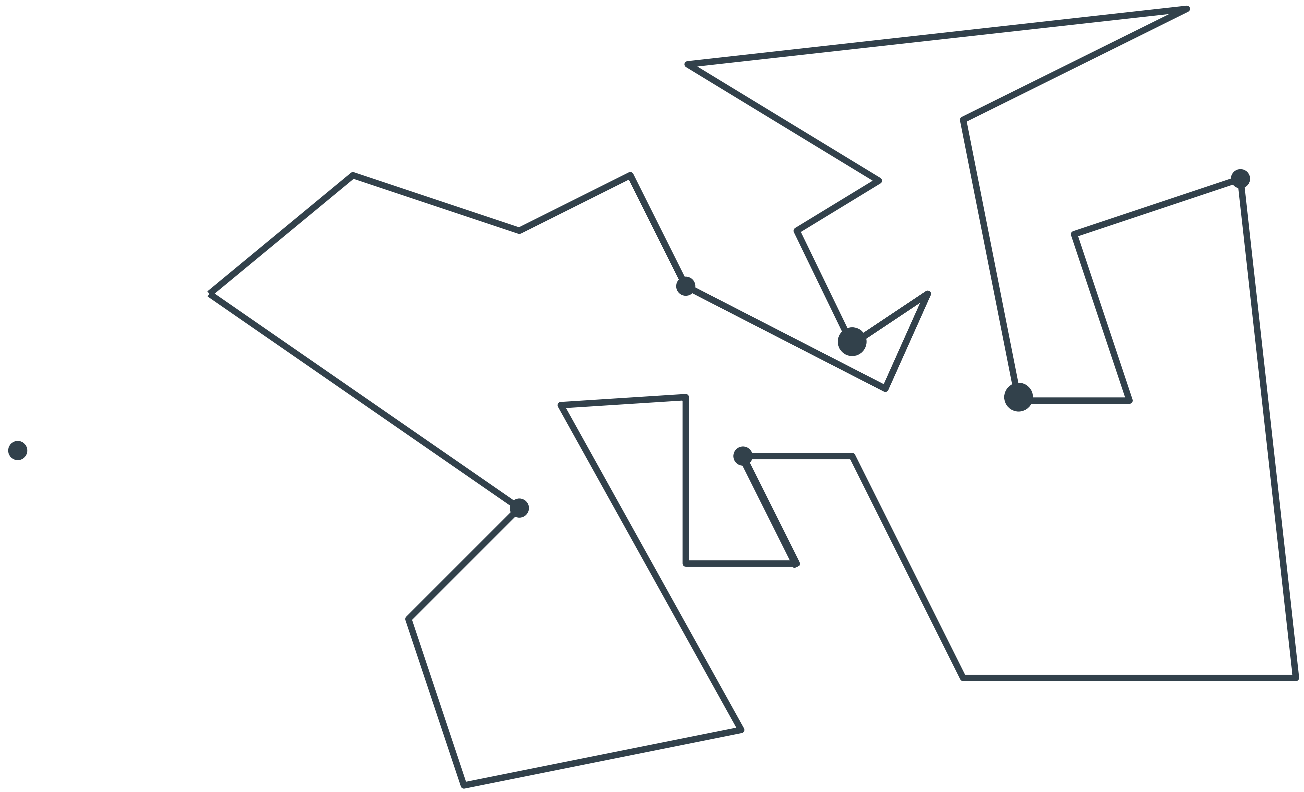
Example



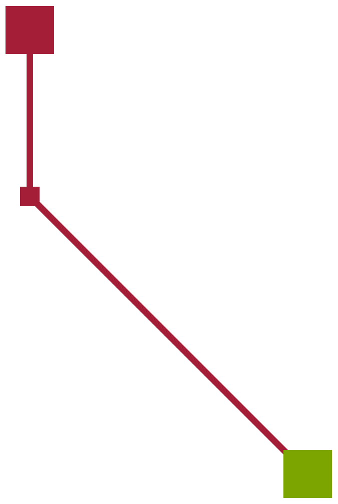
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



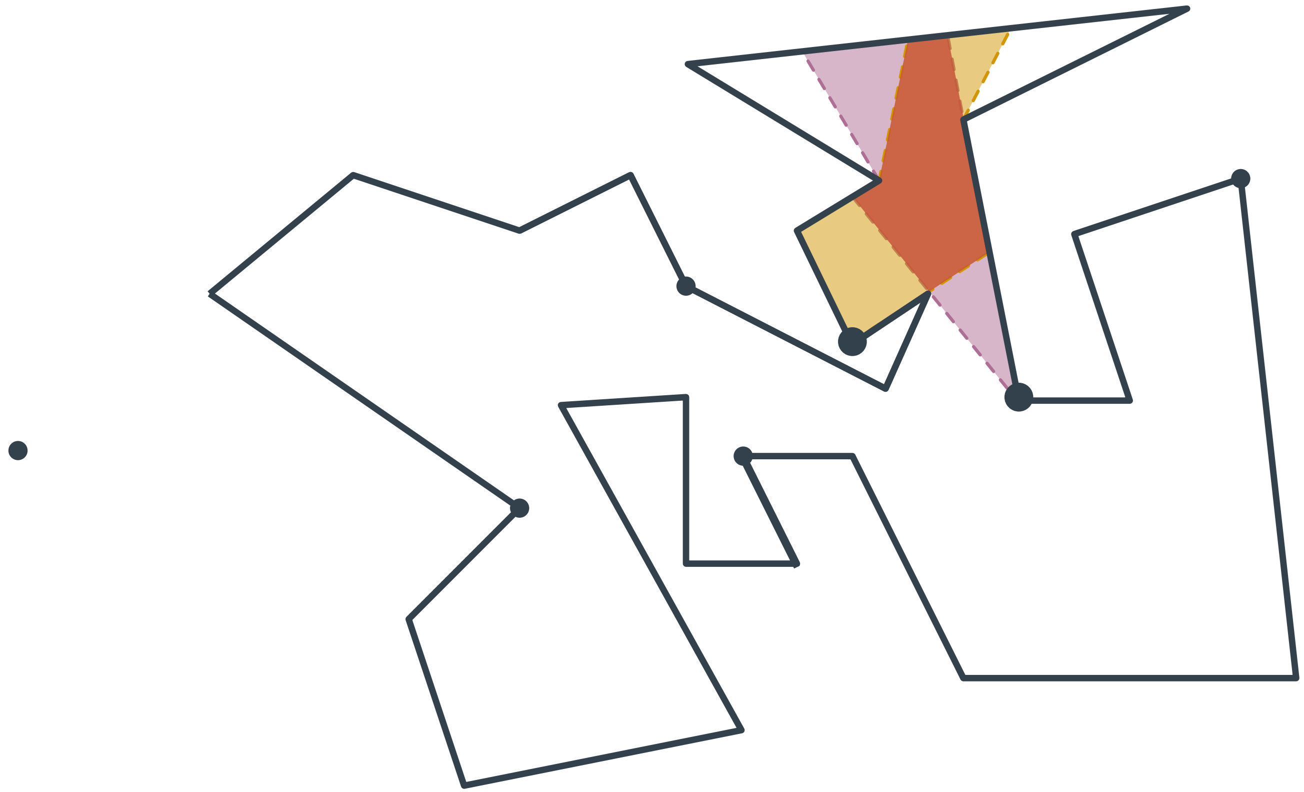
Example



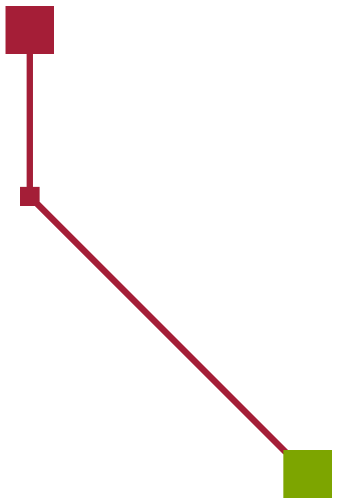
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



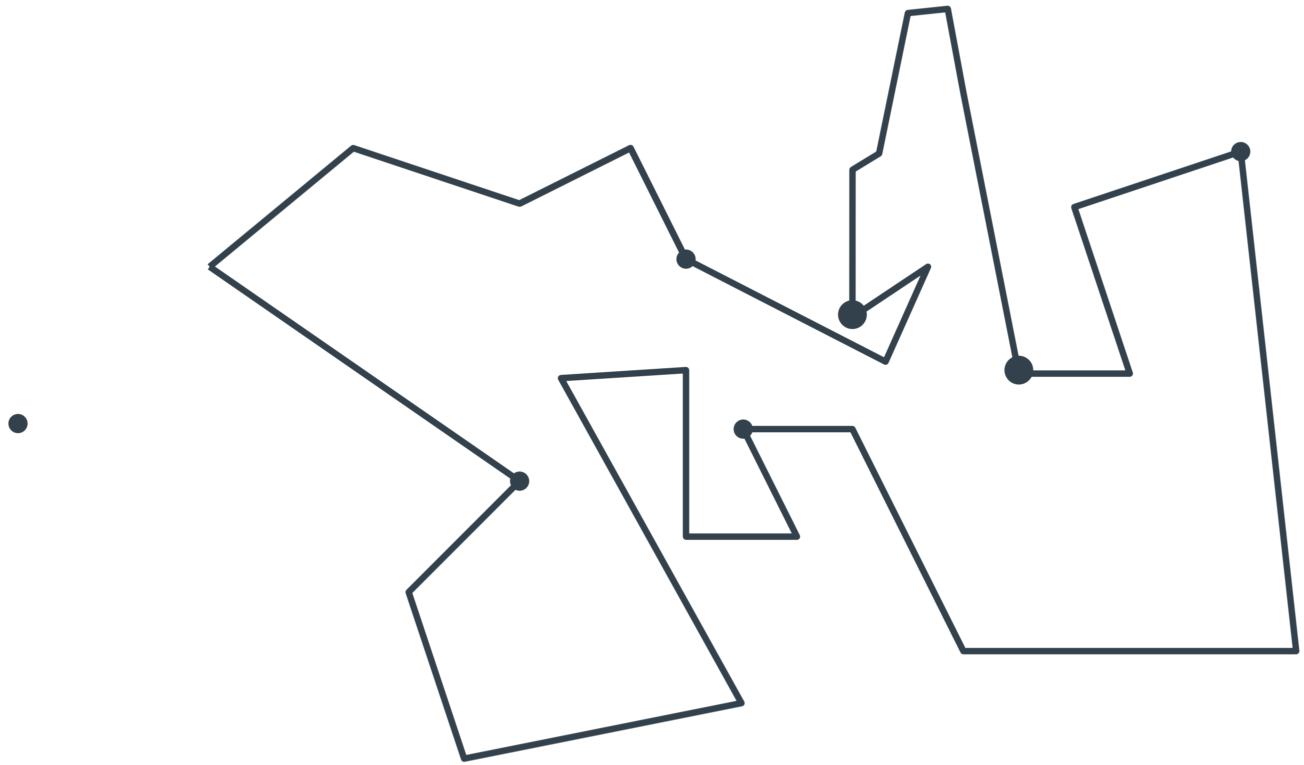
Example



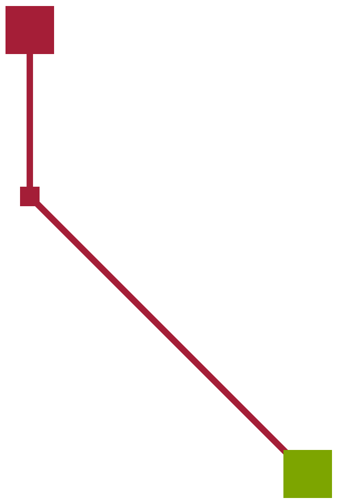
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



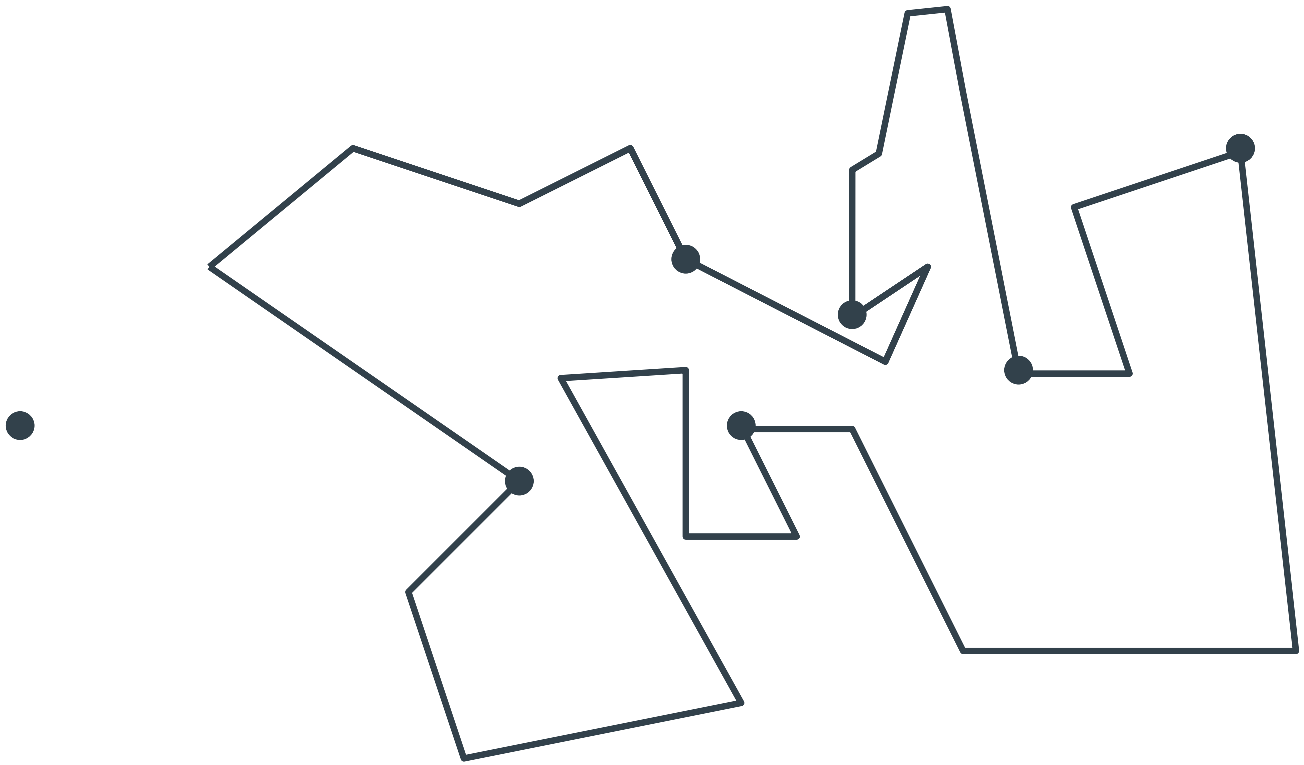
Example



- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons

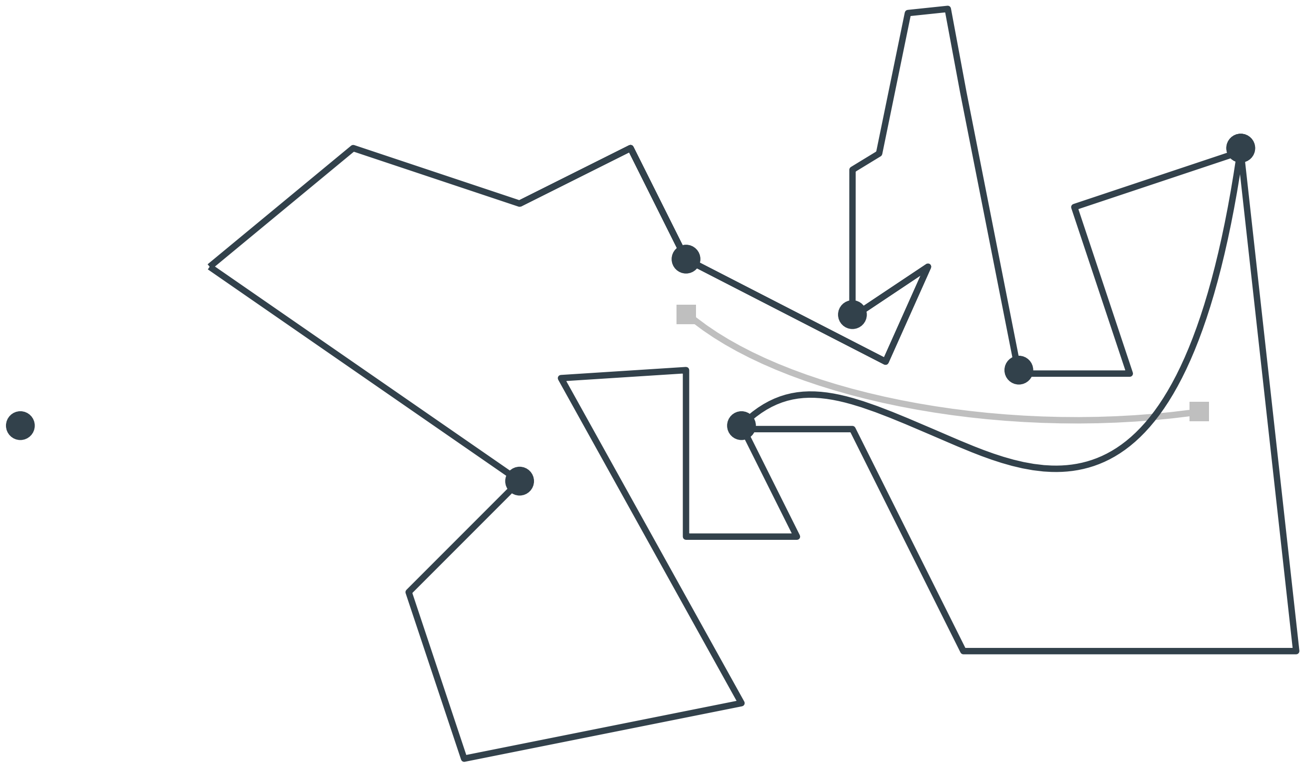


Example



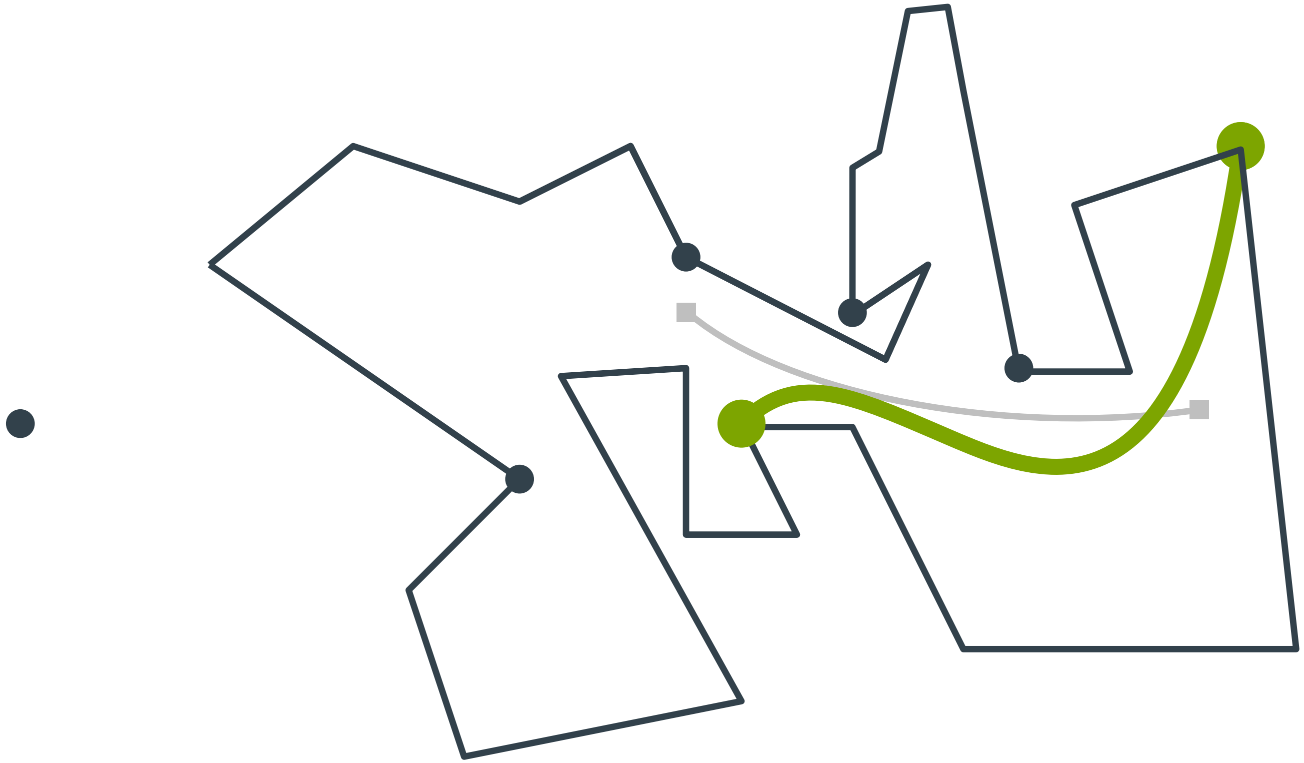
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons

Example



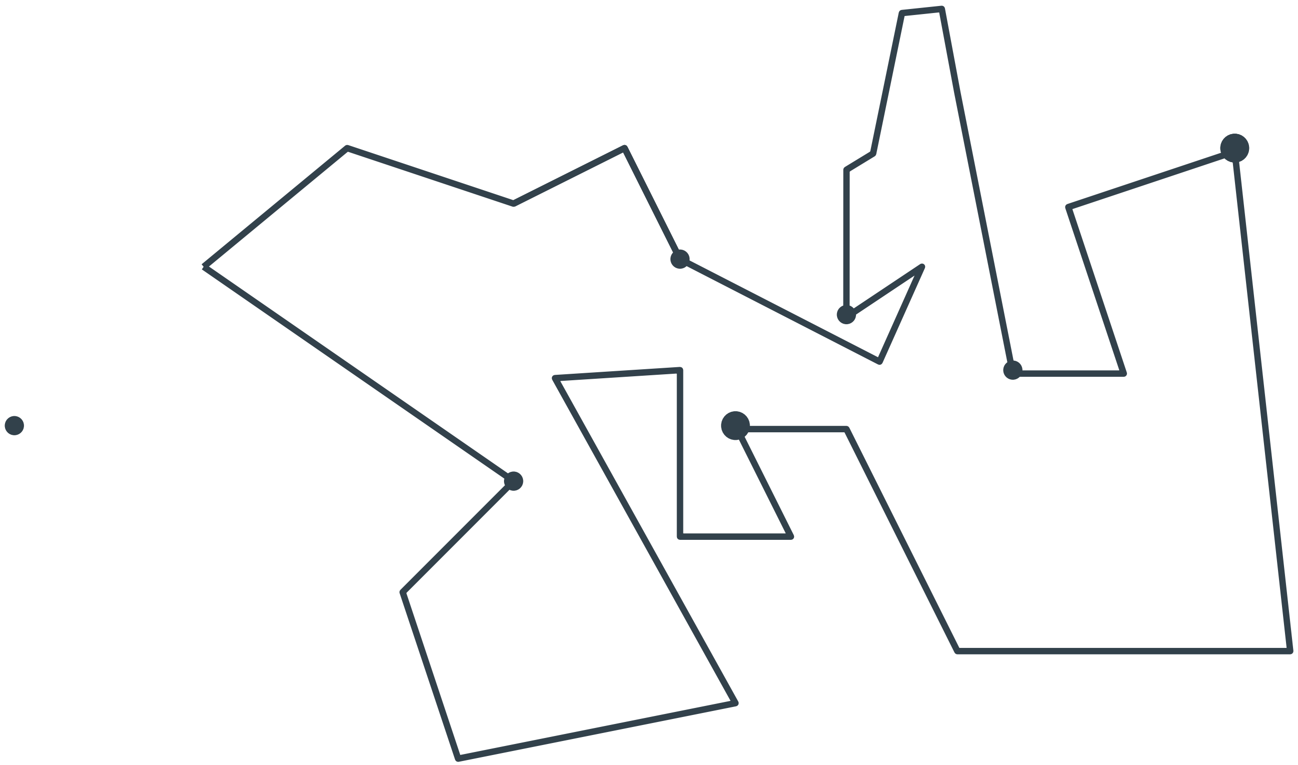
- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons

Example



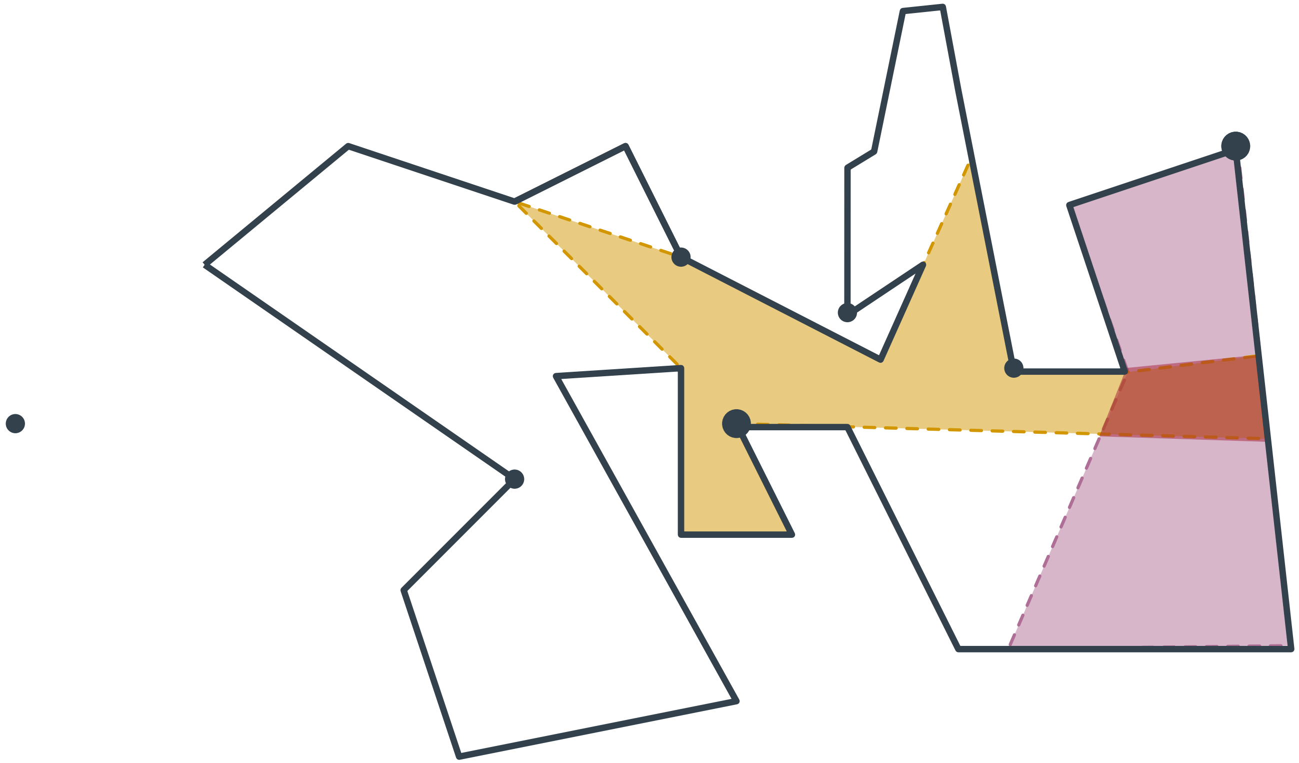
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons

Example



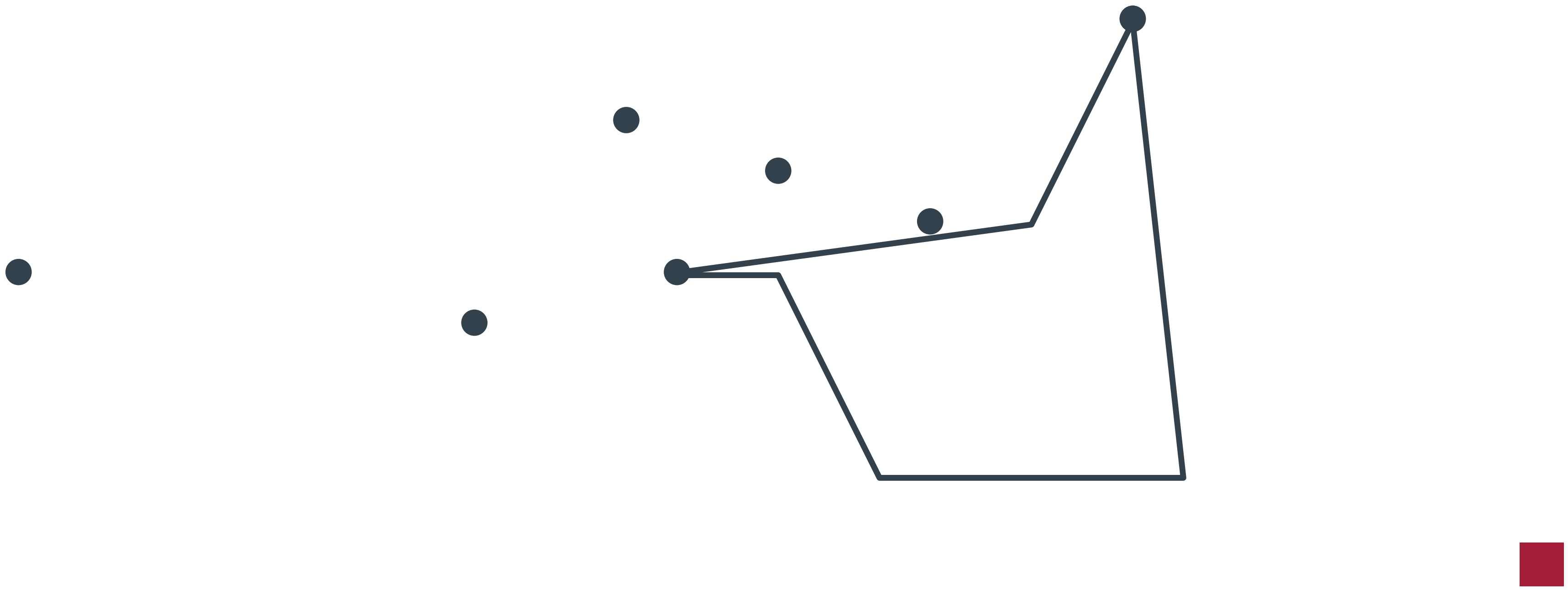
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons

Example



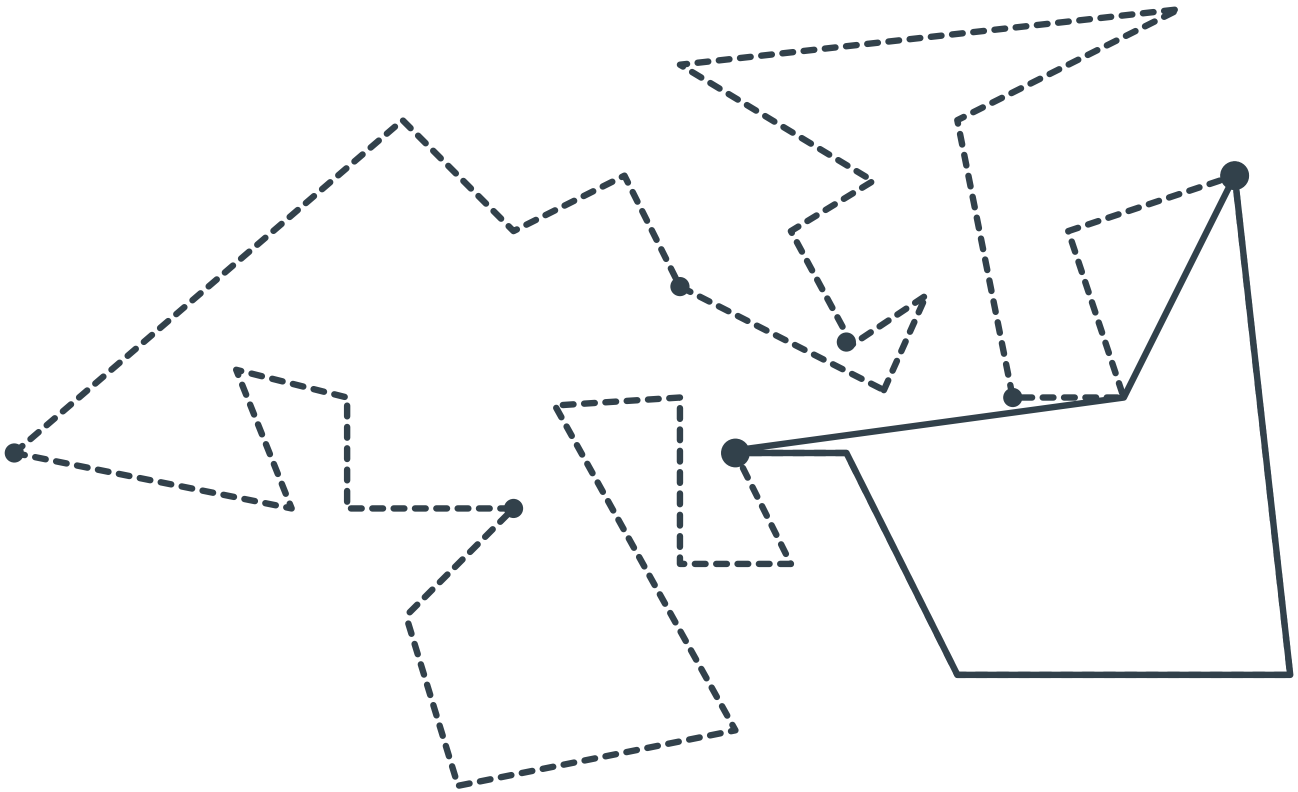
- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges over convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons

Example

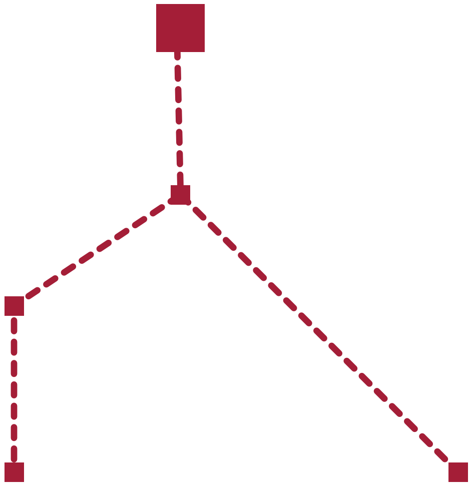


- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons

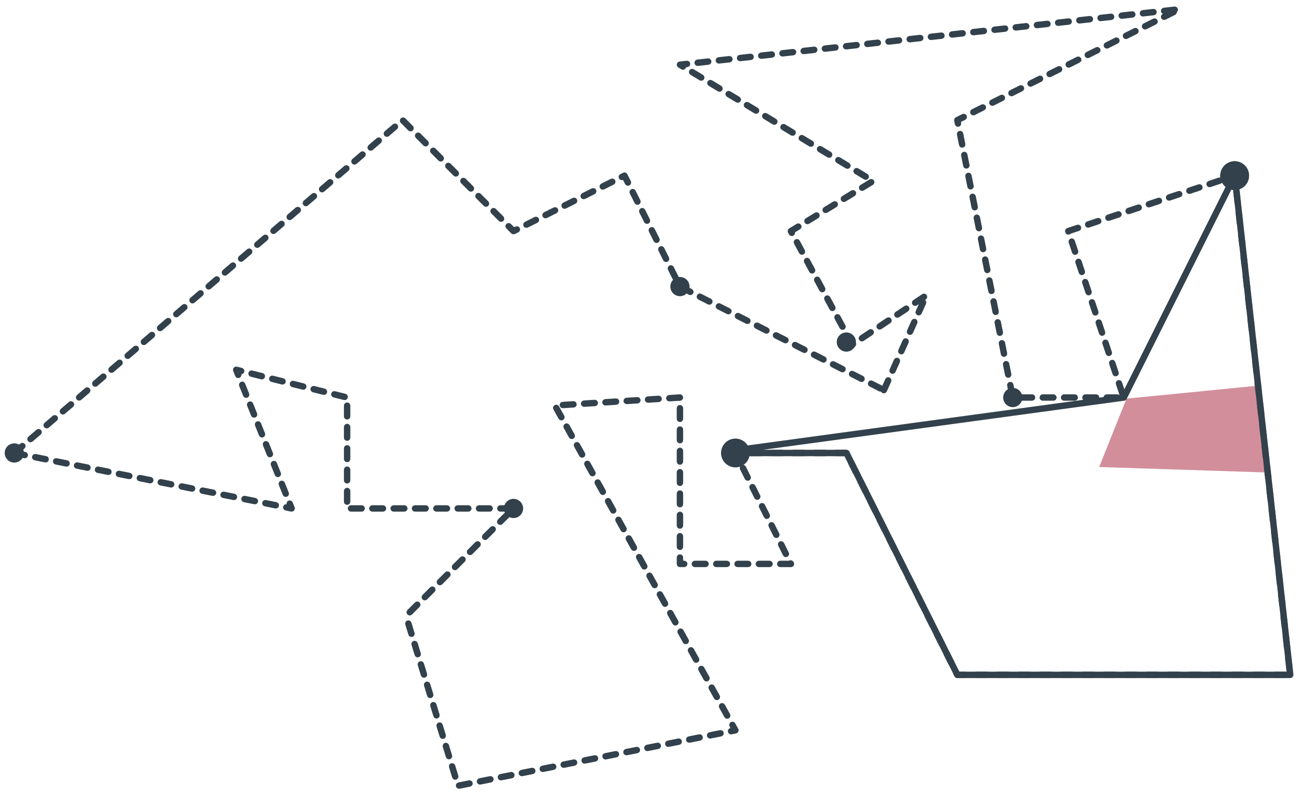
Example



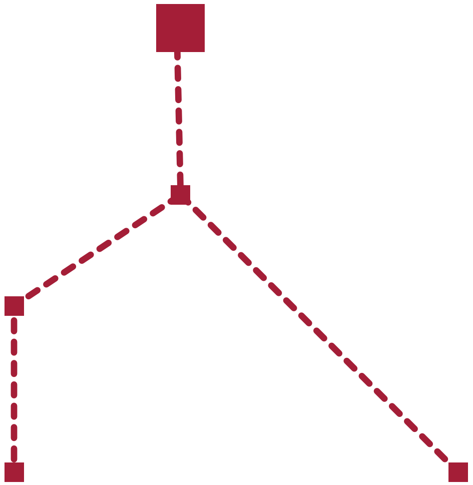
- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



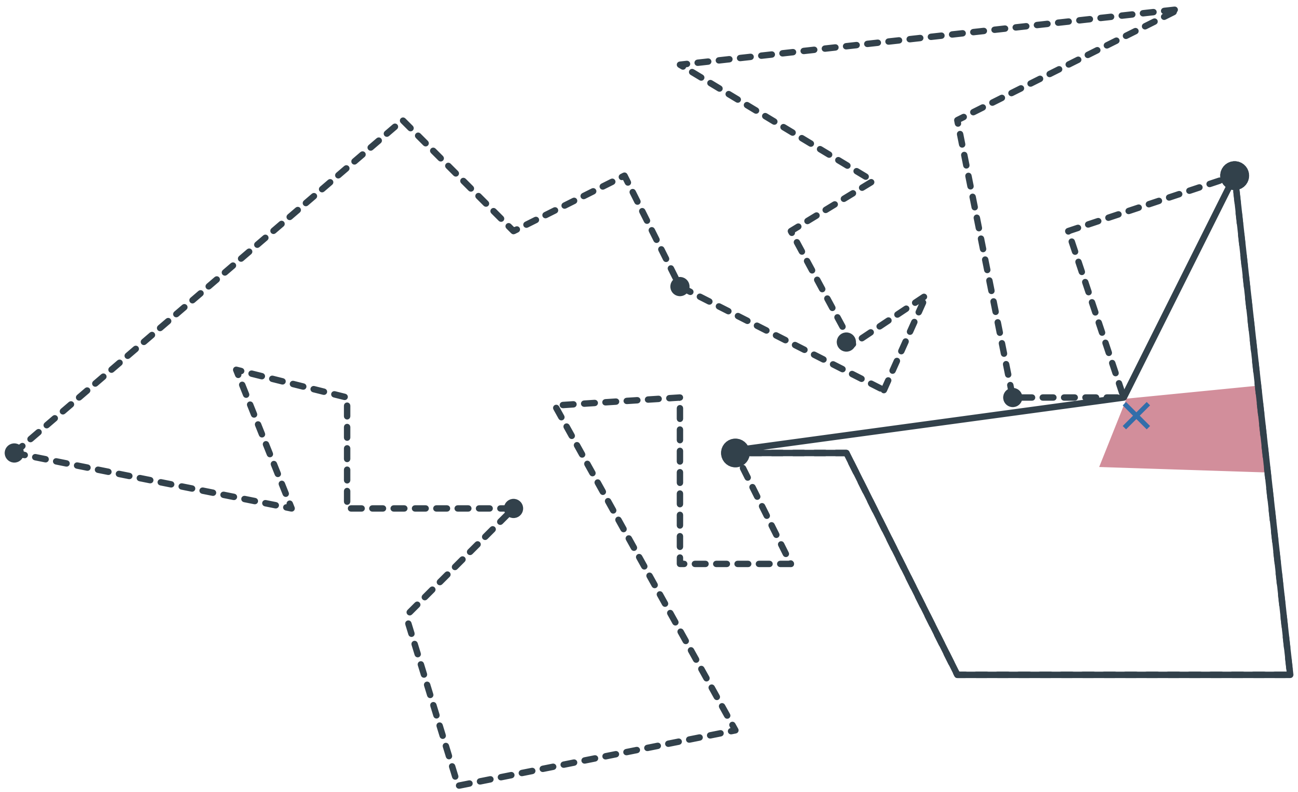
Example



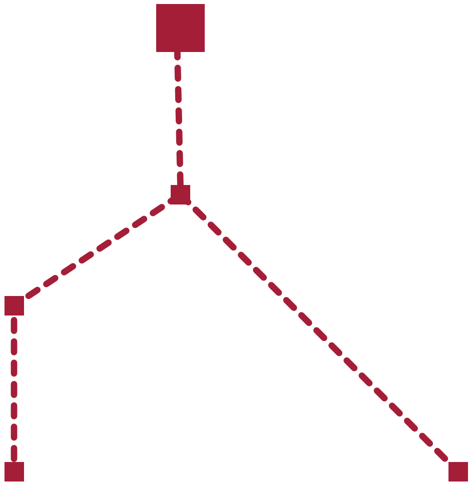
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



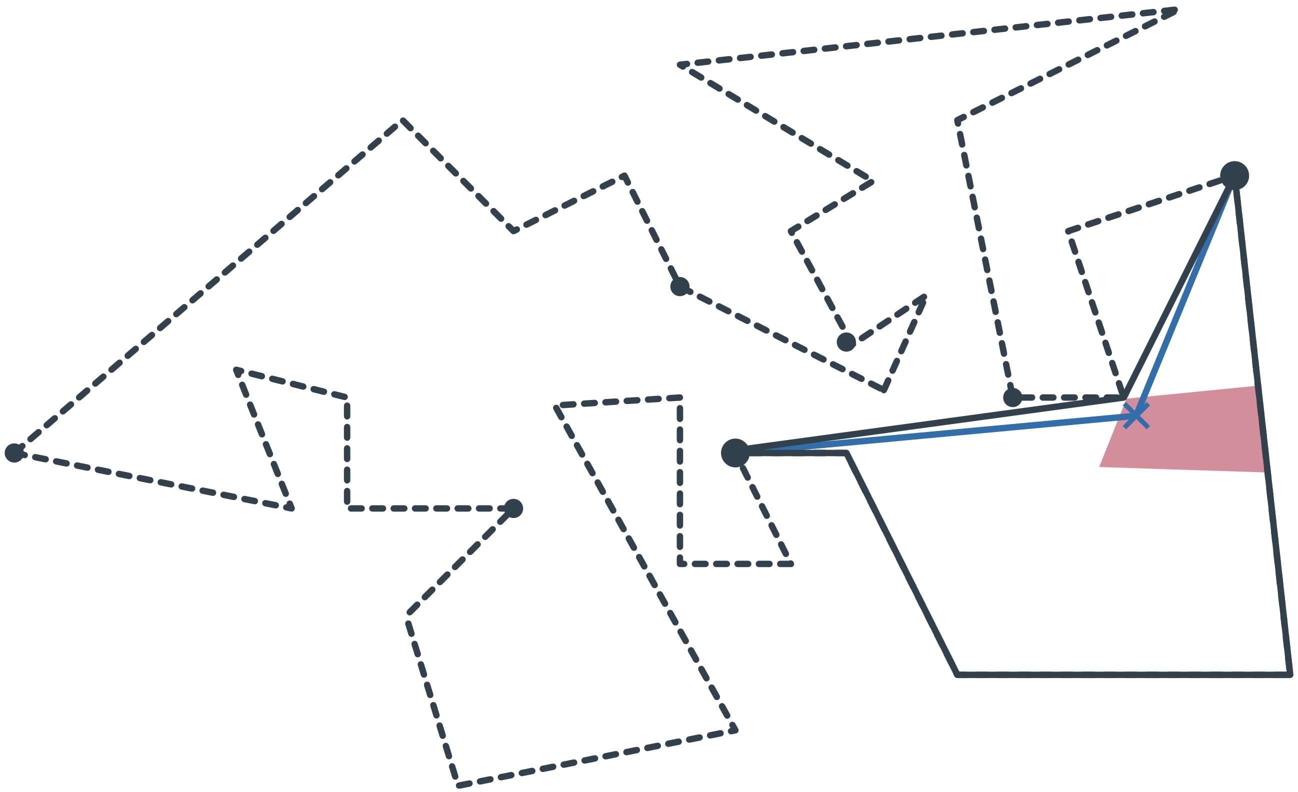
Example

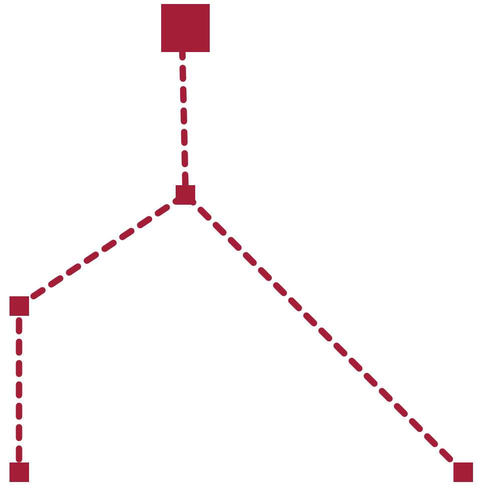


- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



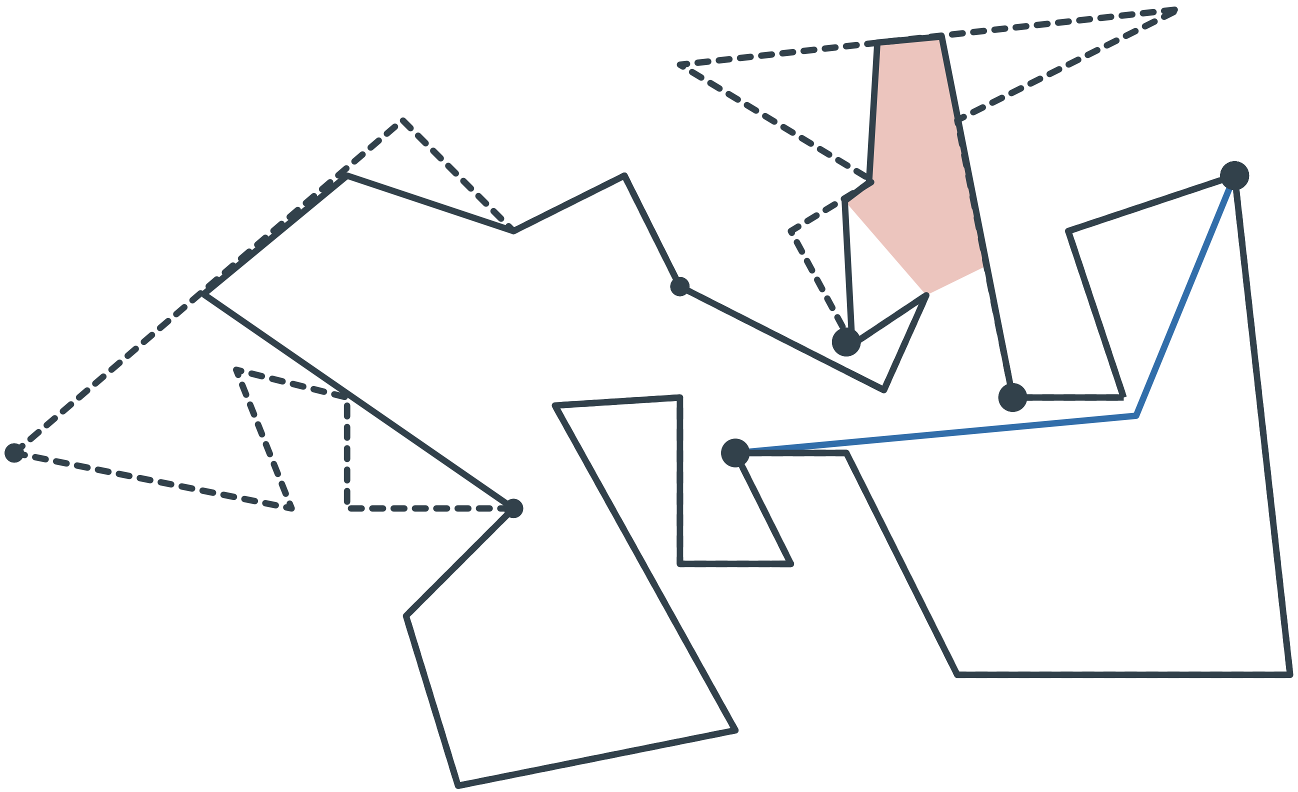
Example



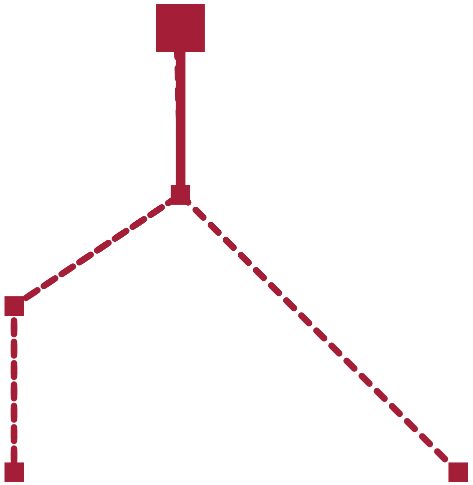


- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons

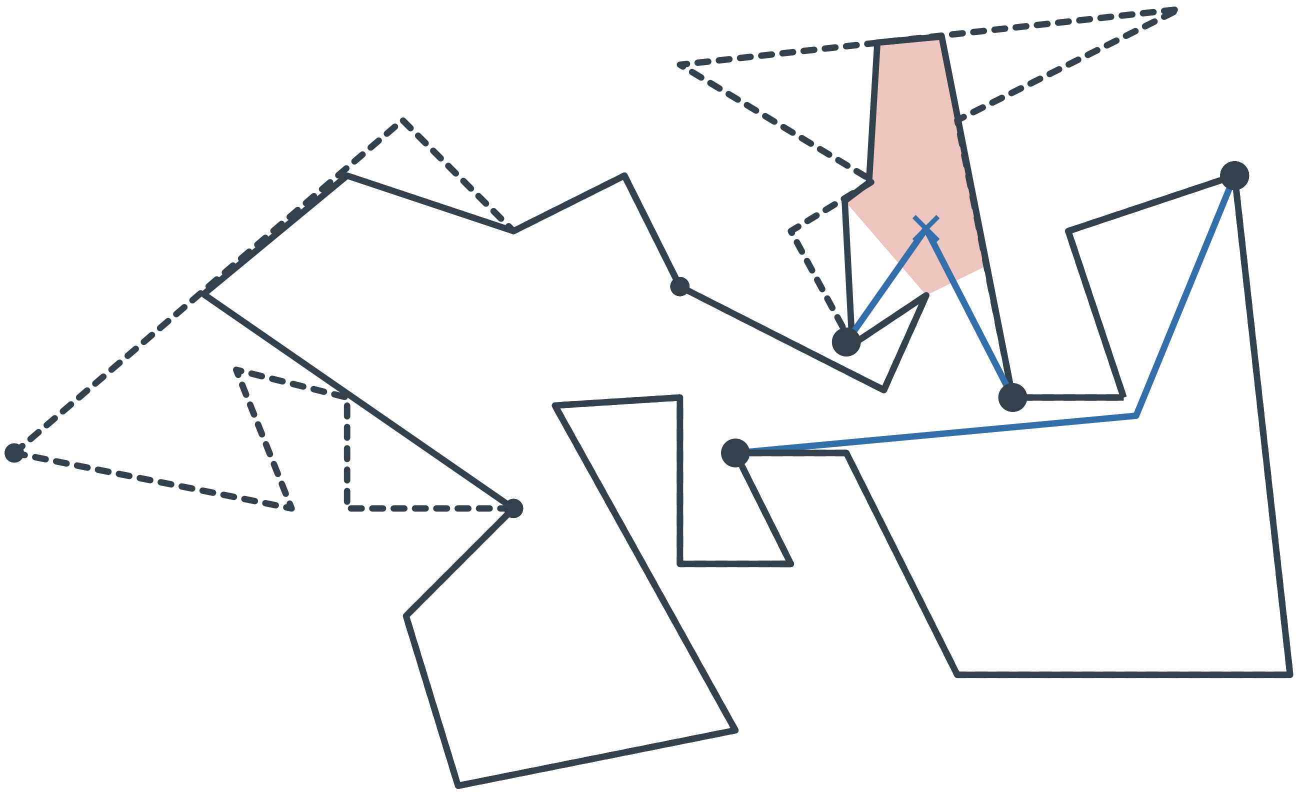
Example



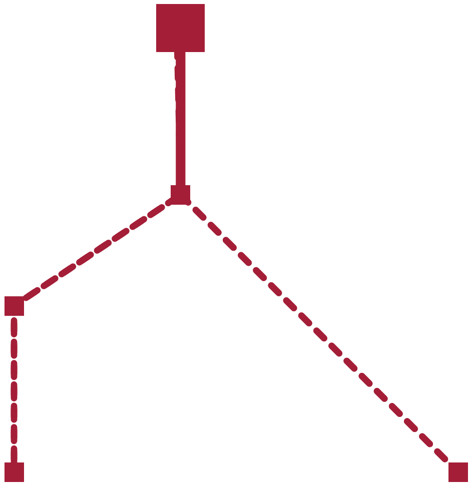
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



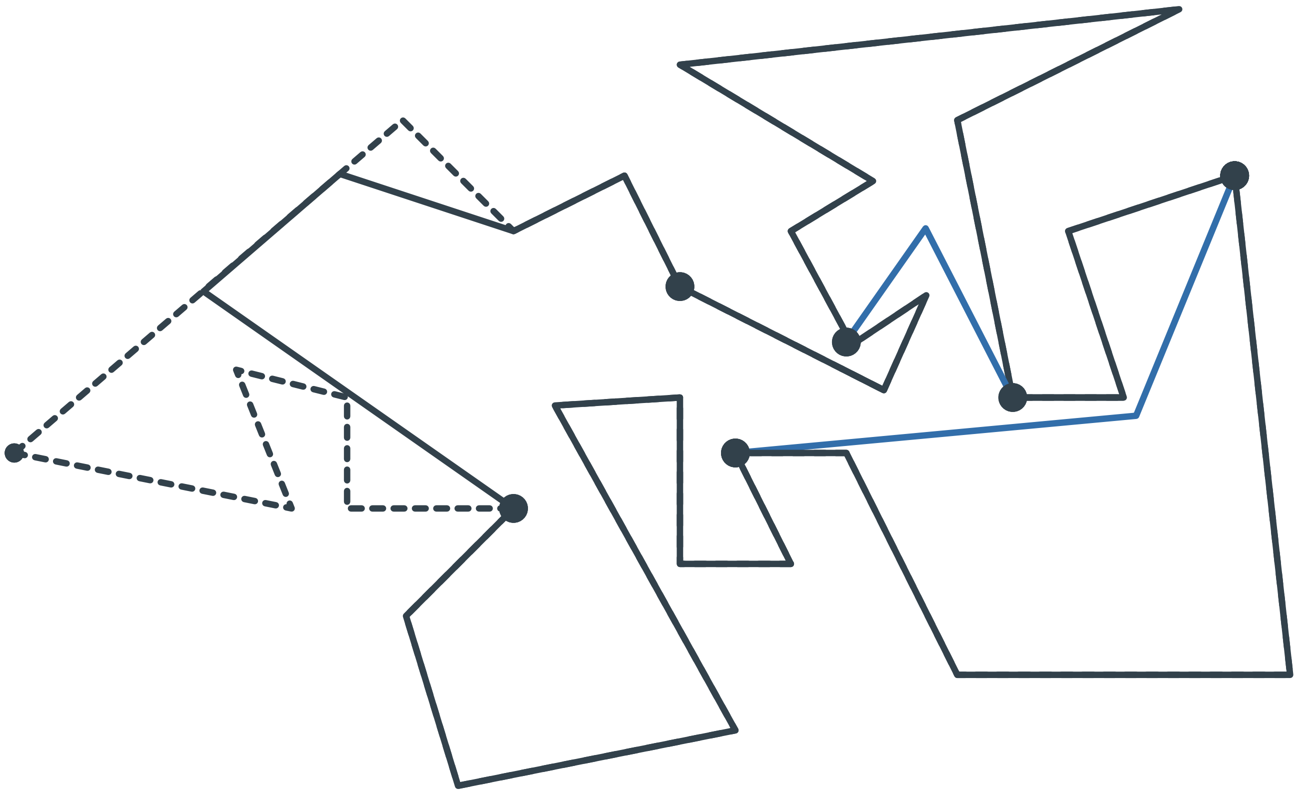
Example



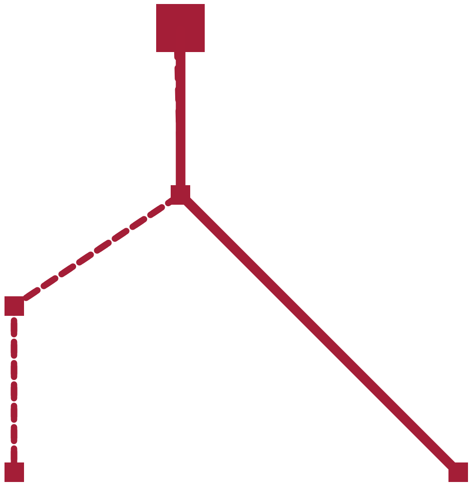
- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



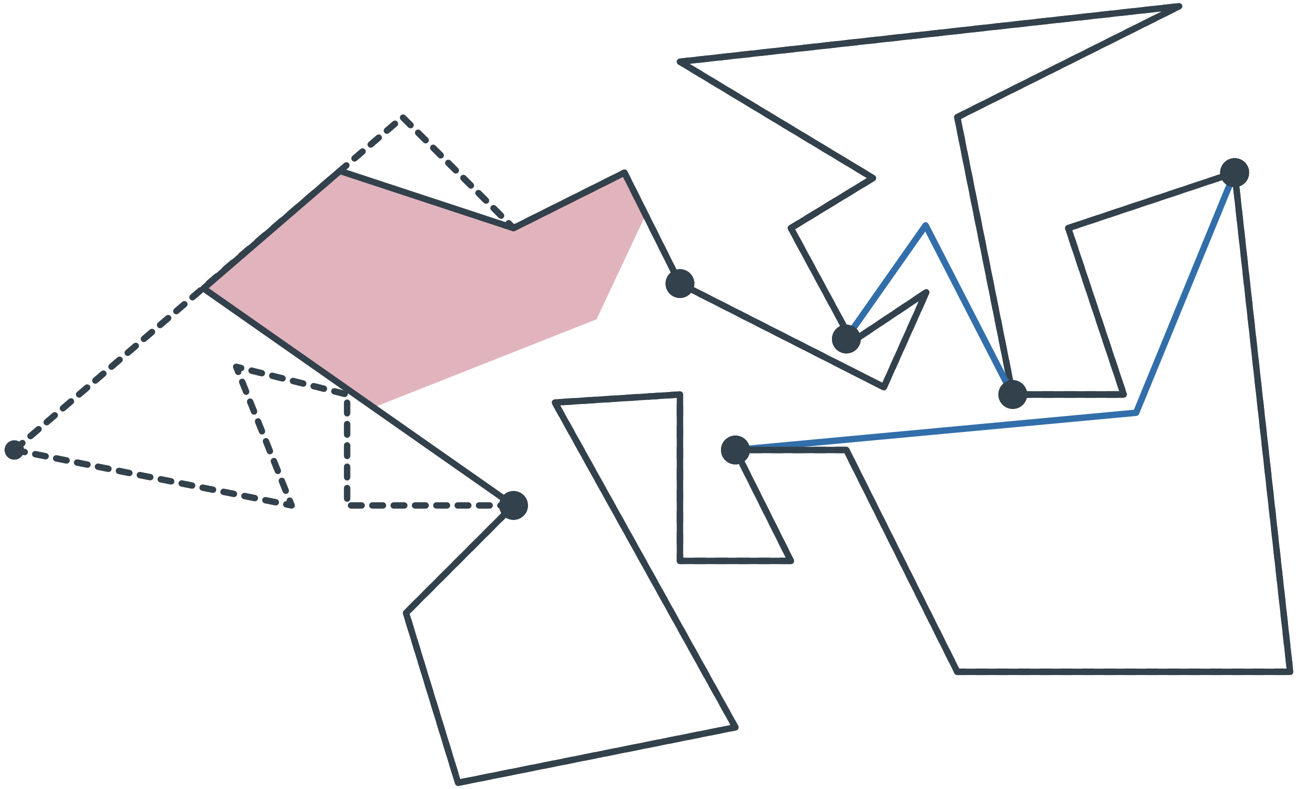
Example



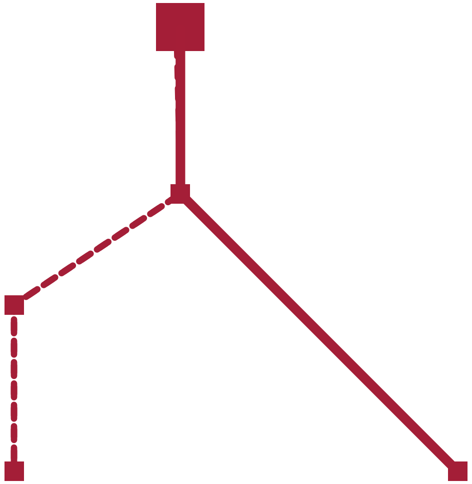
- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



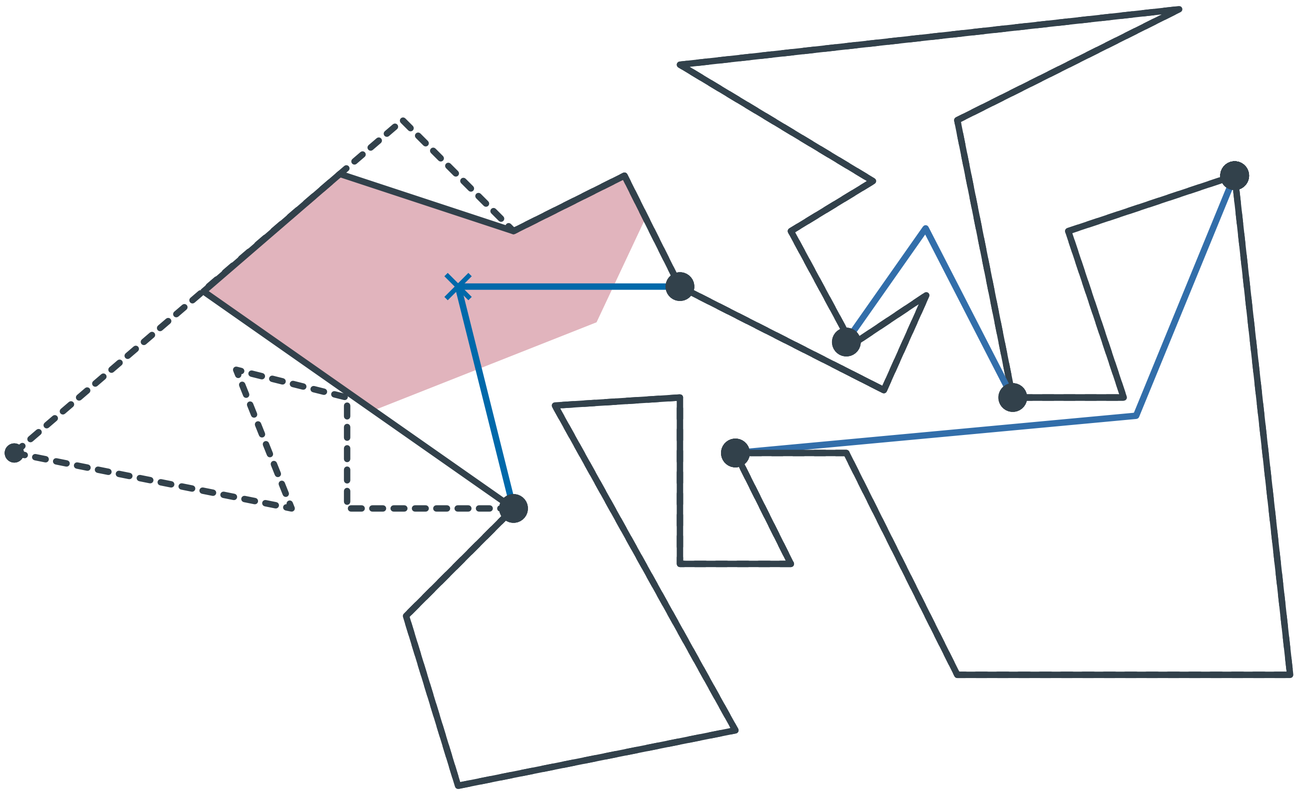
Example



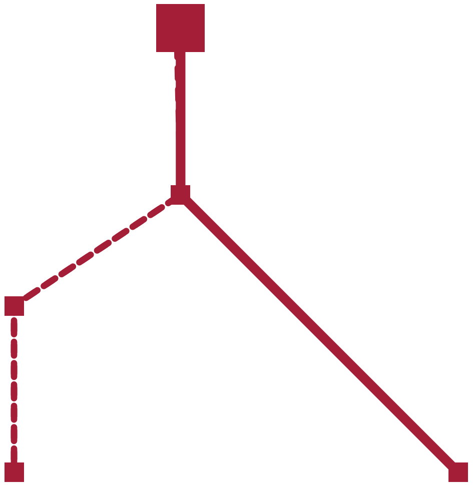
- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



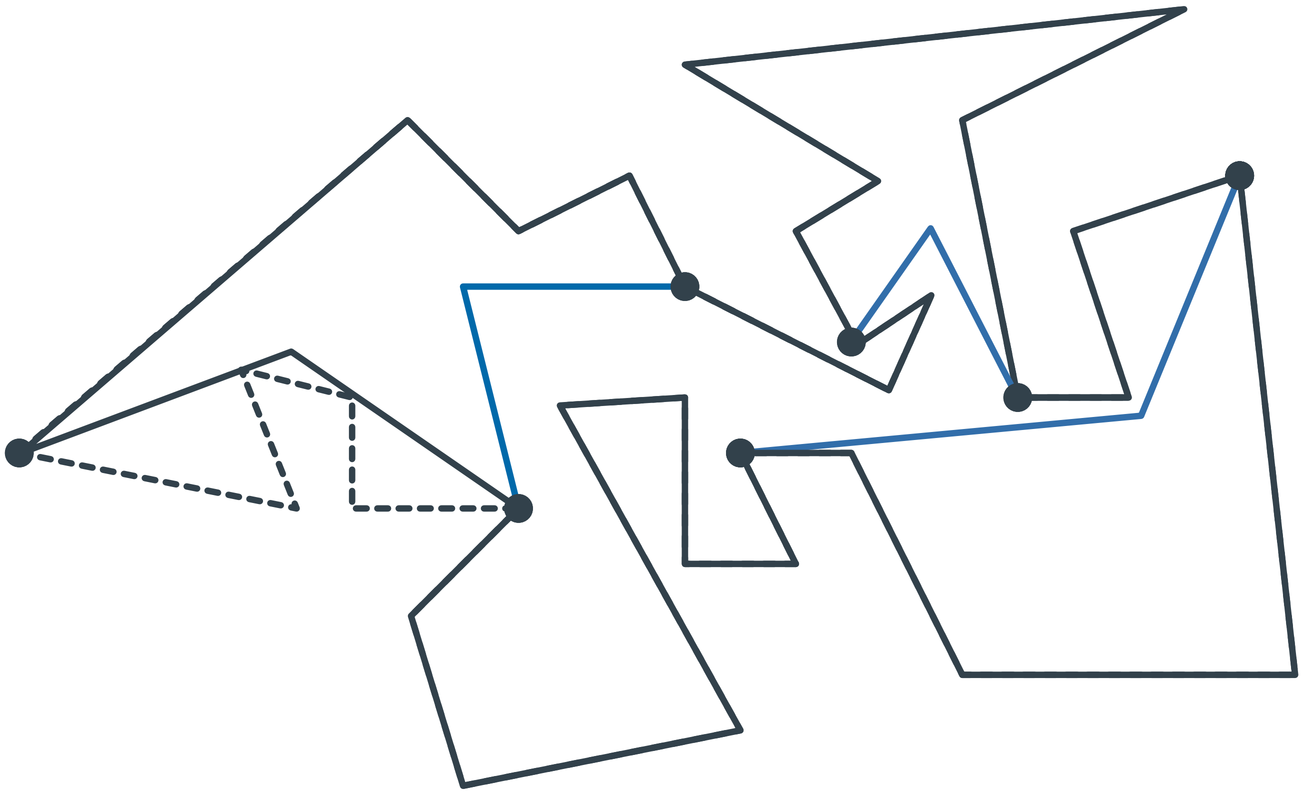
Example



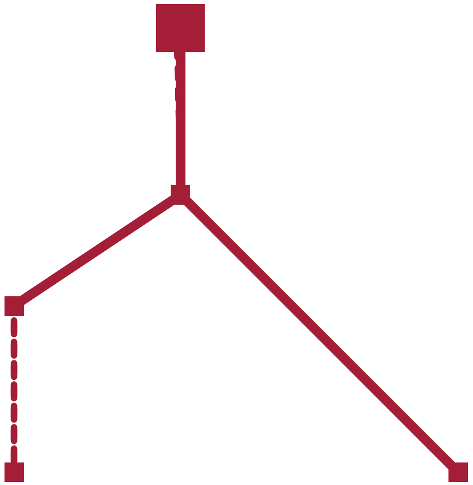
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



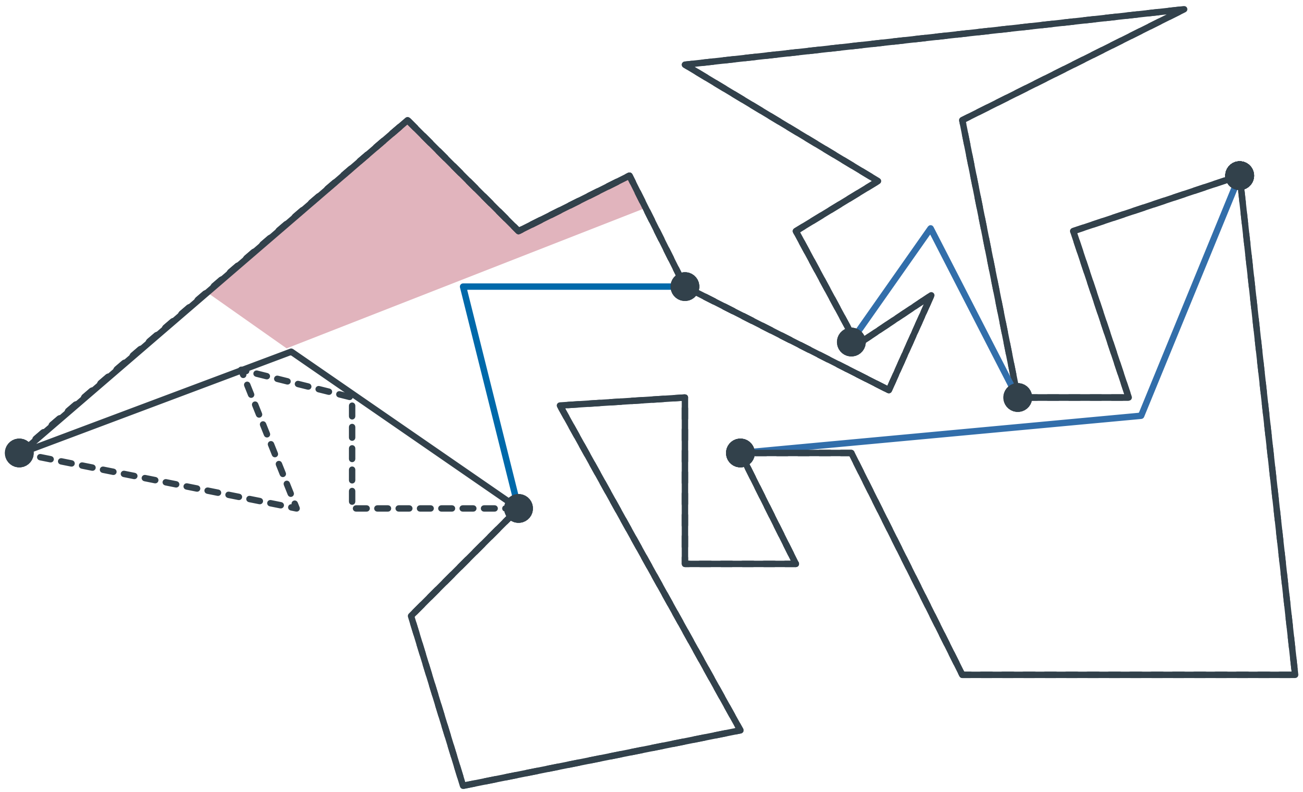
Example



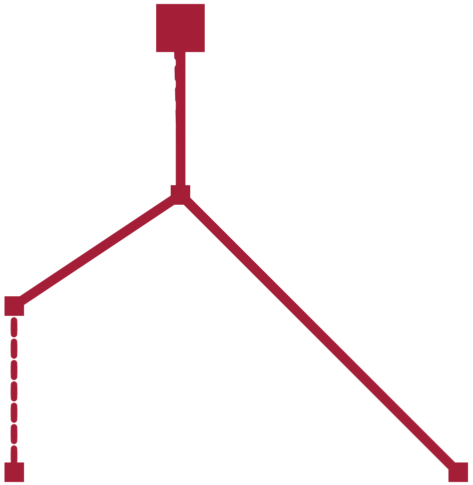
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



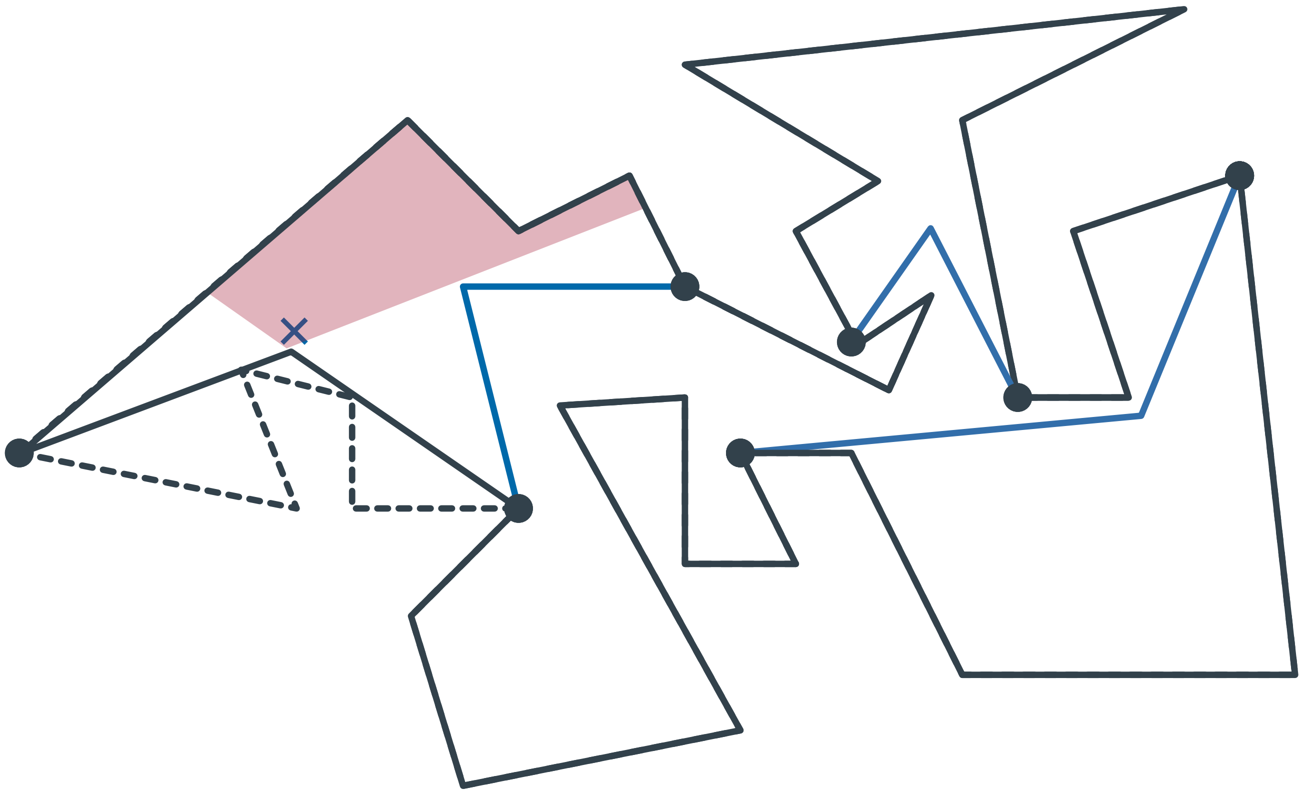
Example



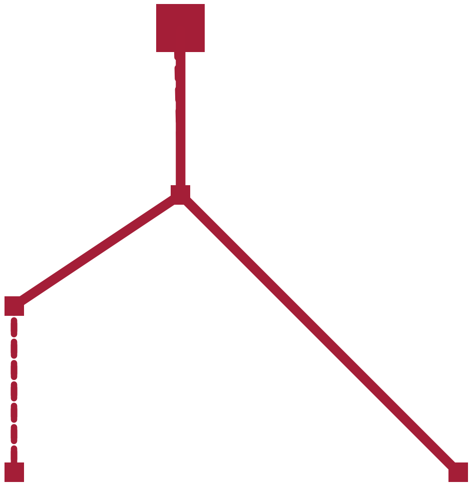
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



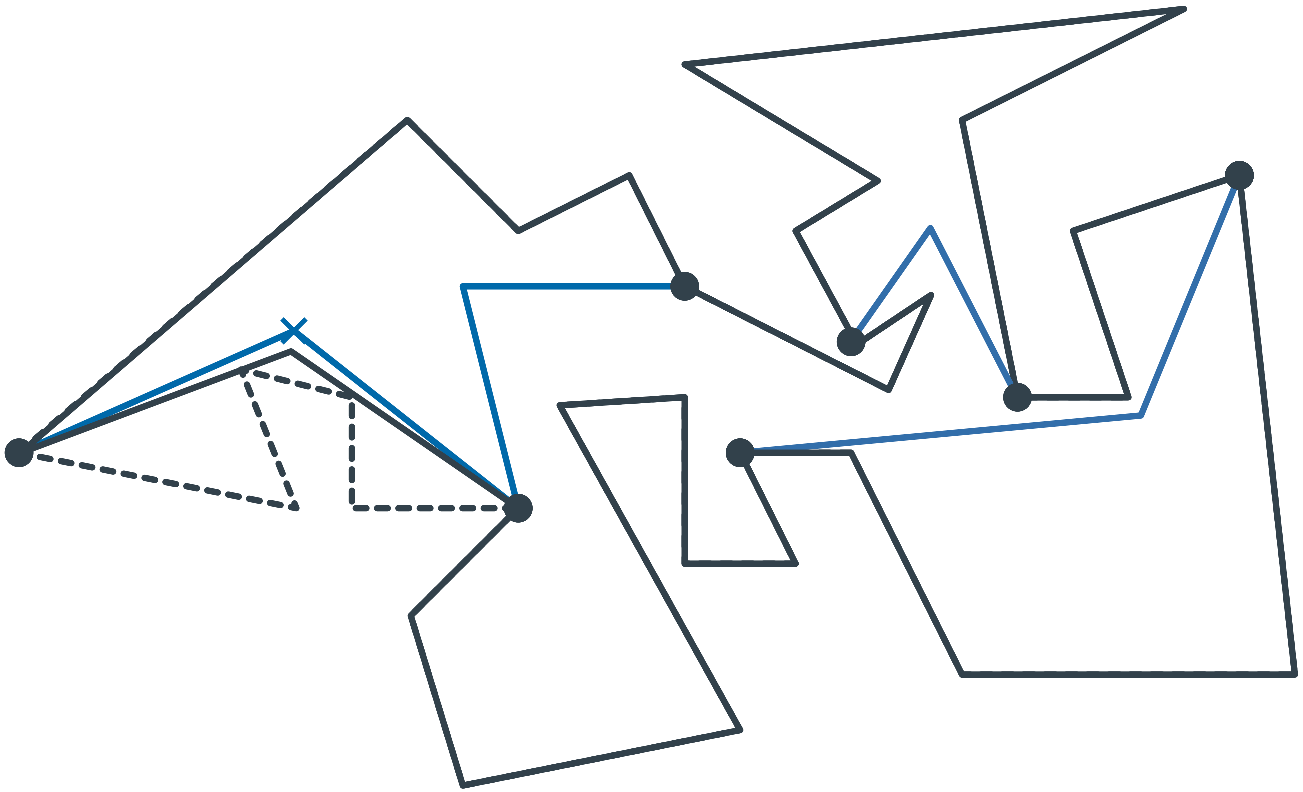
Example



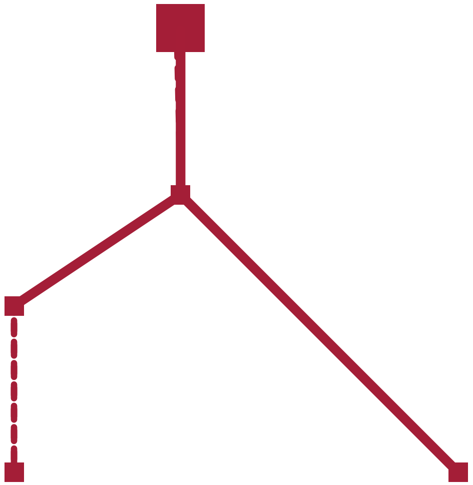
- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



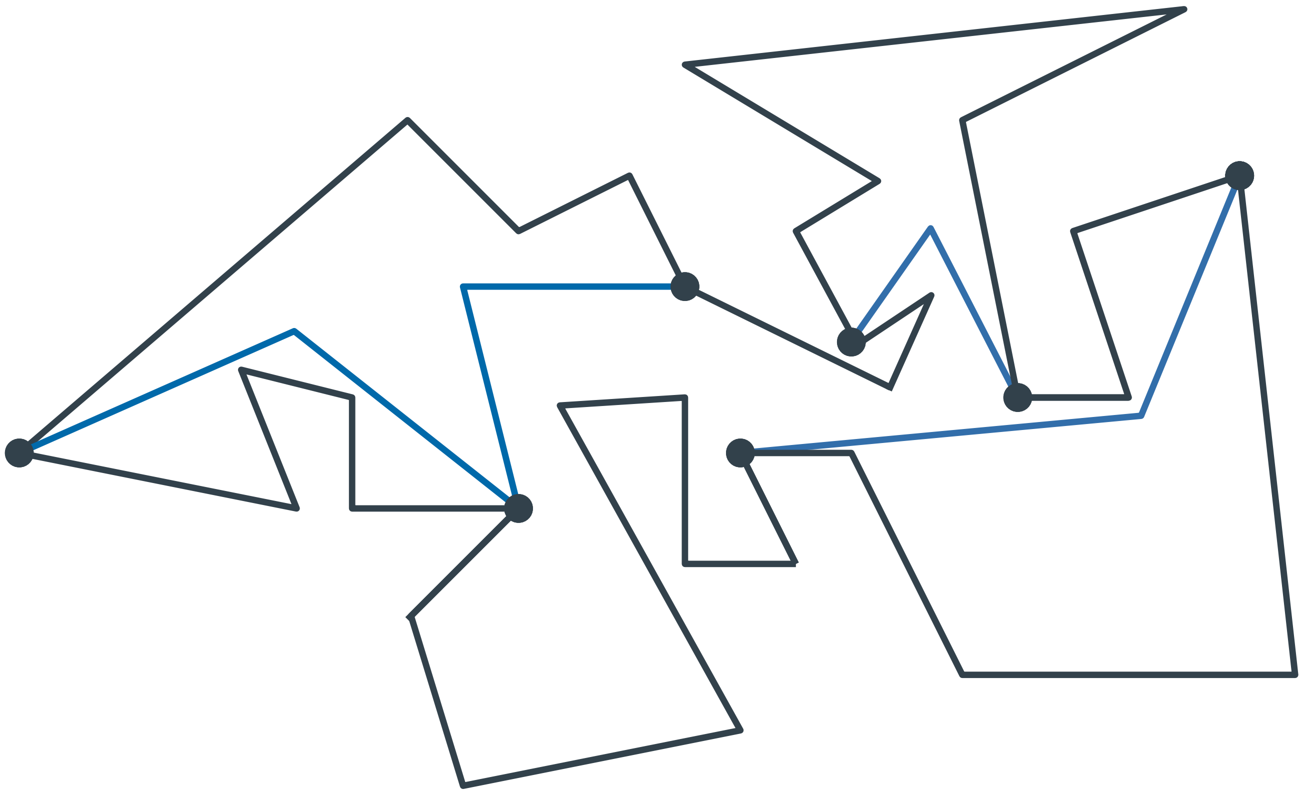
Example



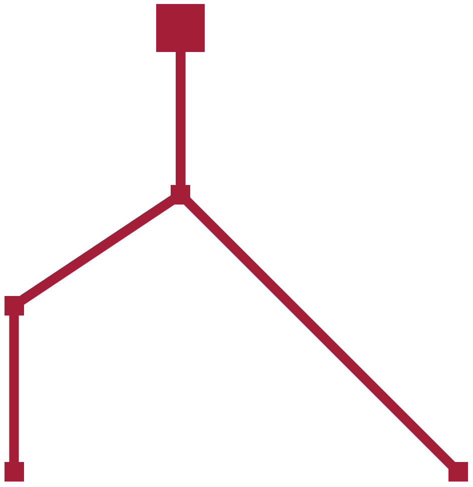
- Traverse the dual tree twice
- Bottom-up: refine the polygon
  - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



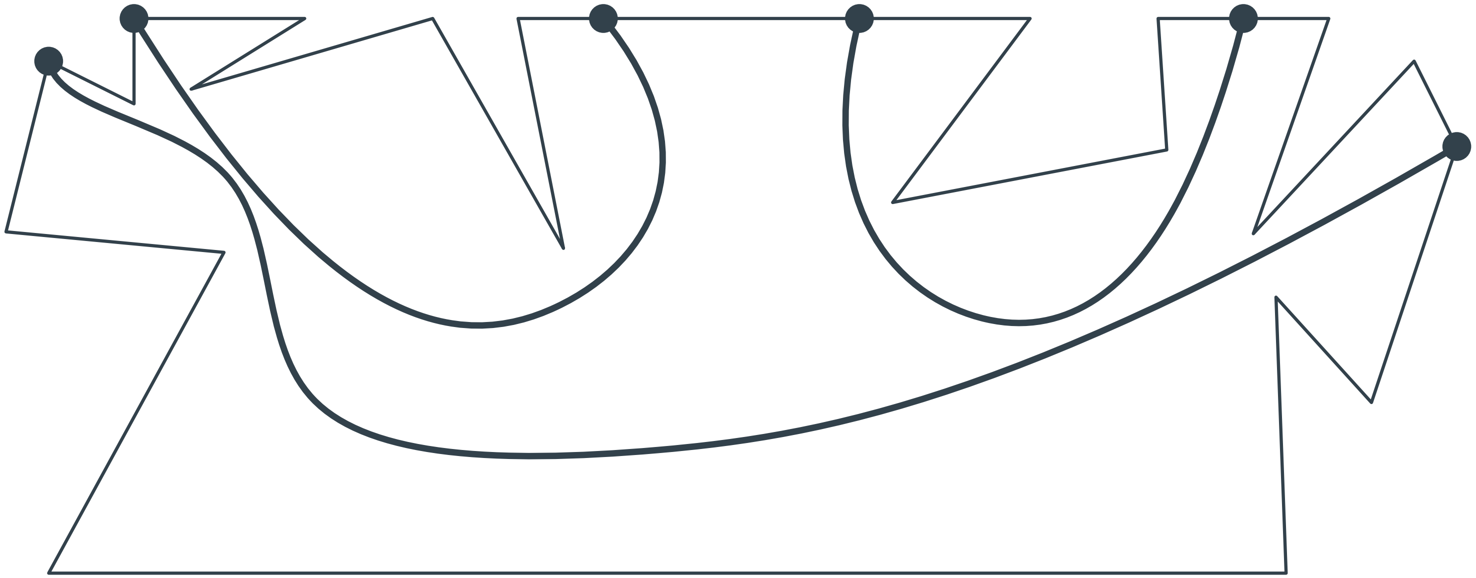
Example



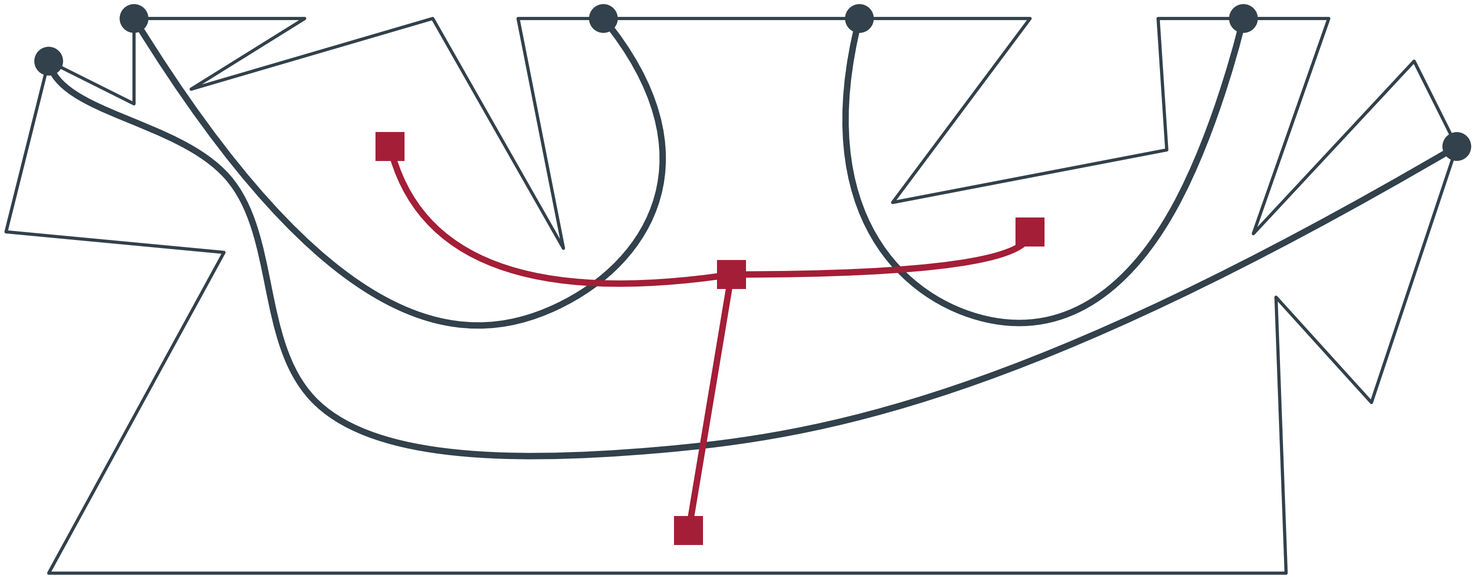
- Traverse the dual tree twice
- Bottom-up: refine the polygon
    - choose reflex edges before convex edges (corresponding to leaves)
- Top-down: place bend points in the refined polygons



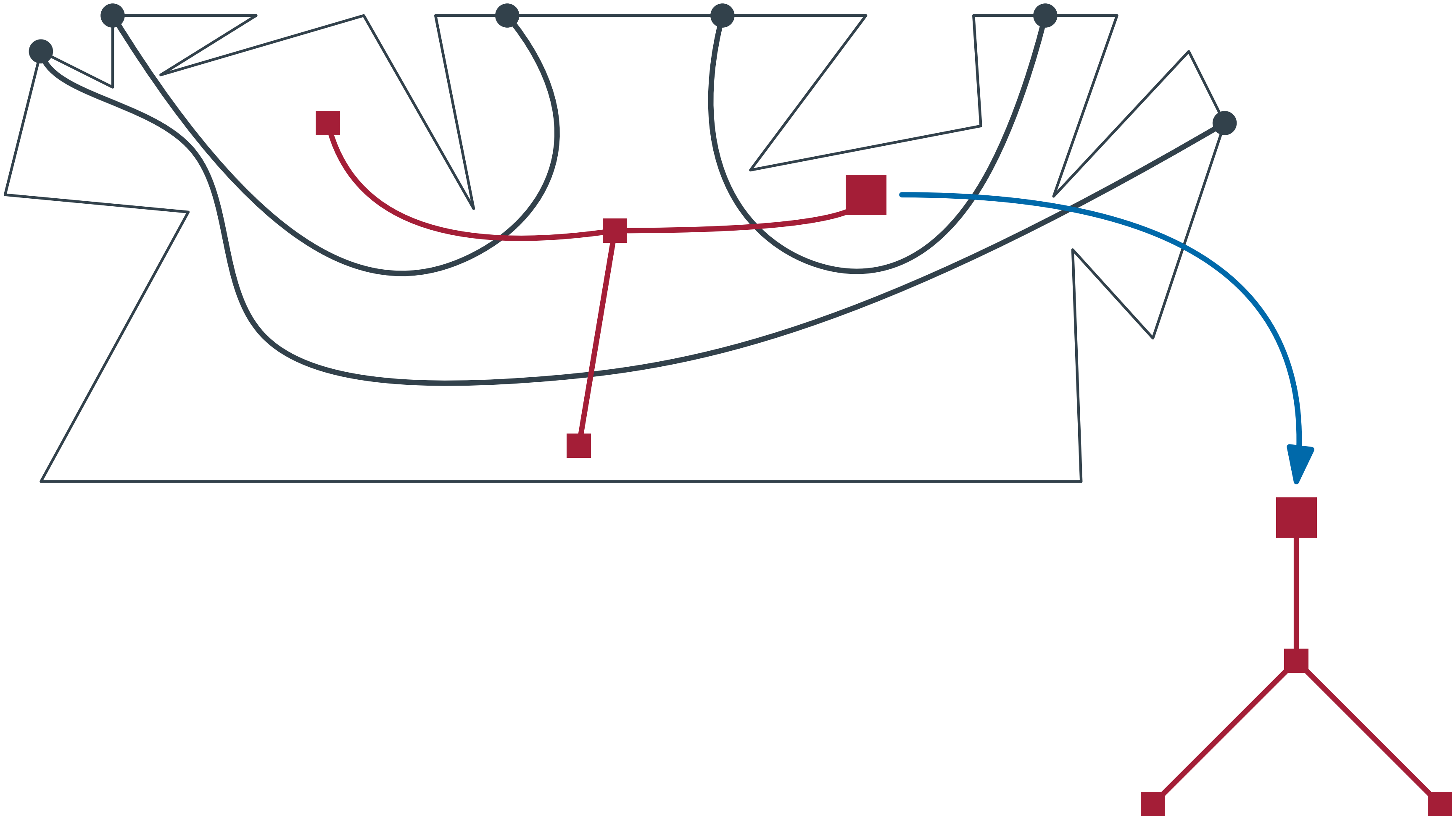
Why this order?



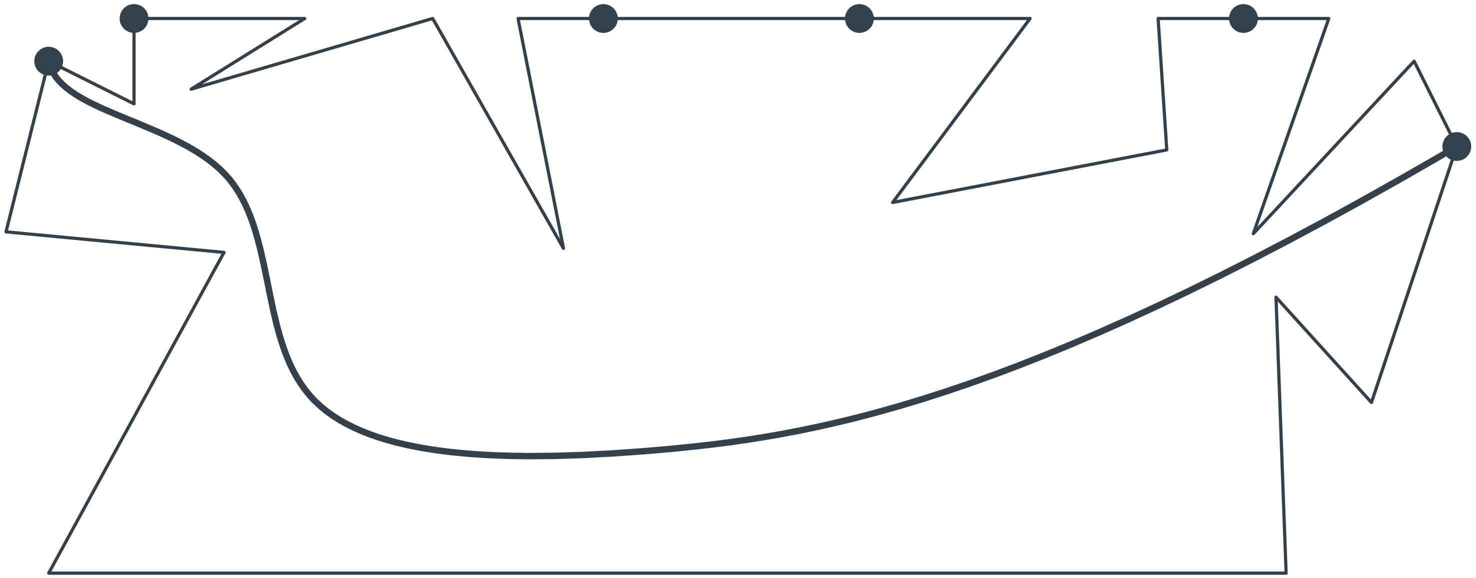
Why this order?



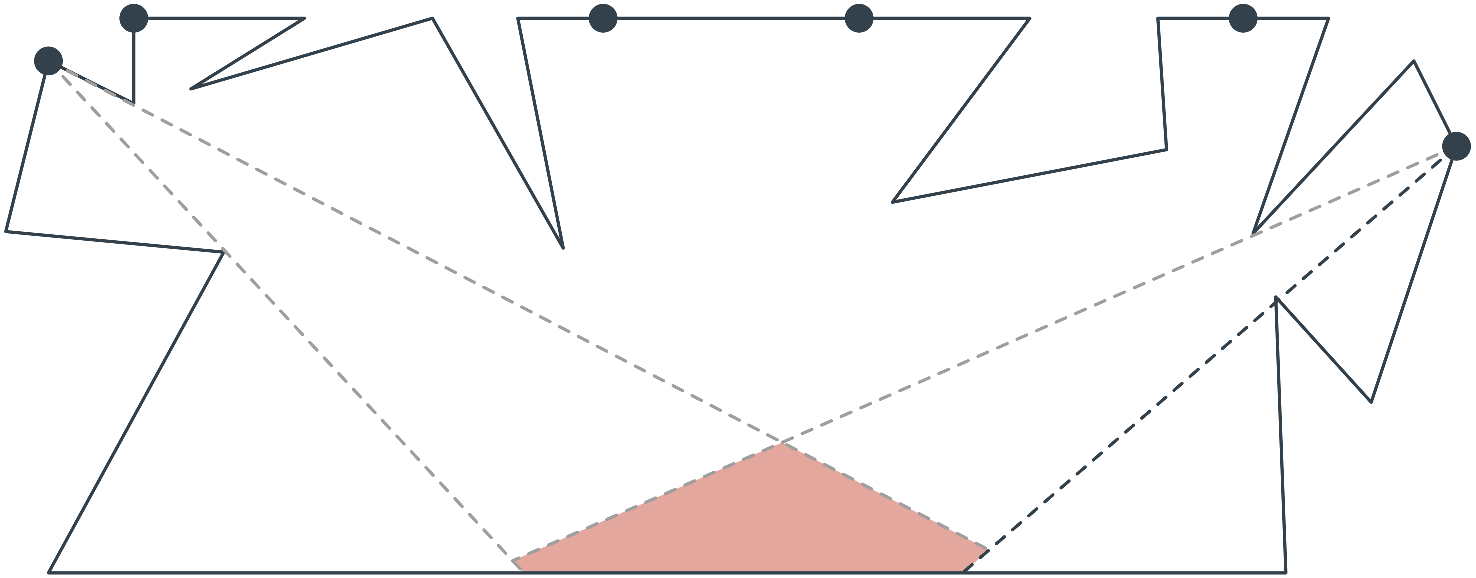
Why this order?



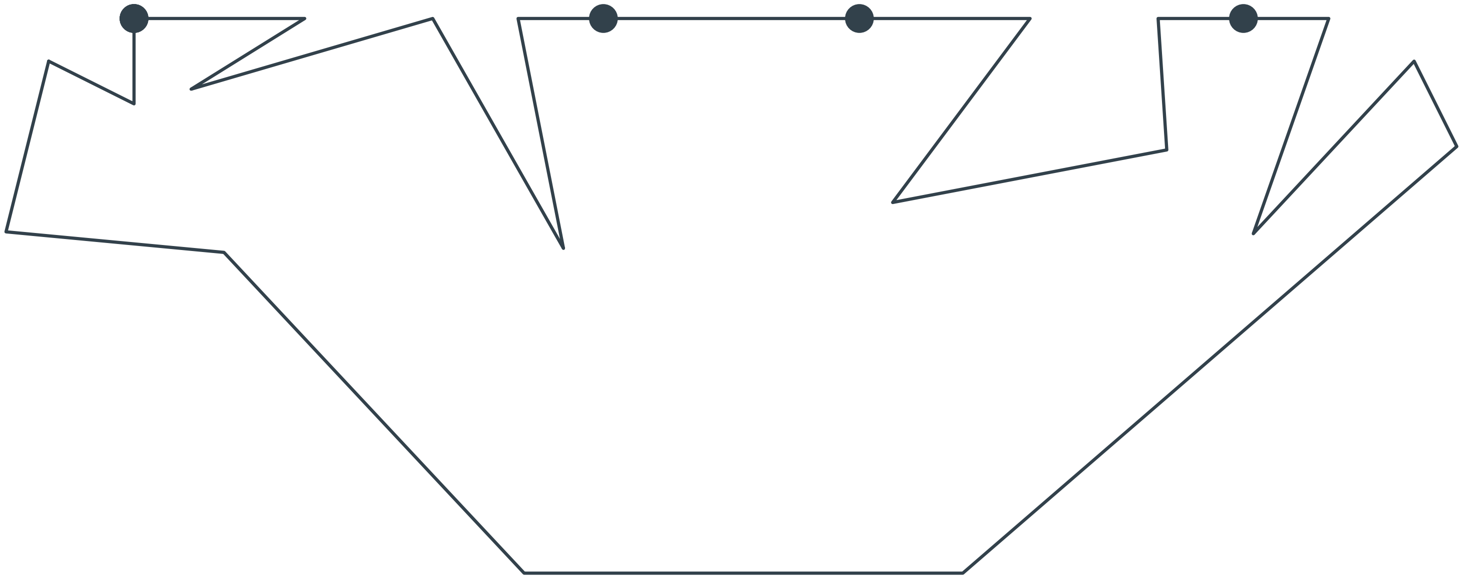
Why this order?



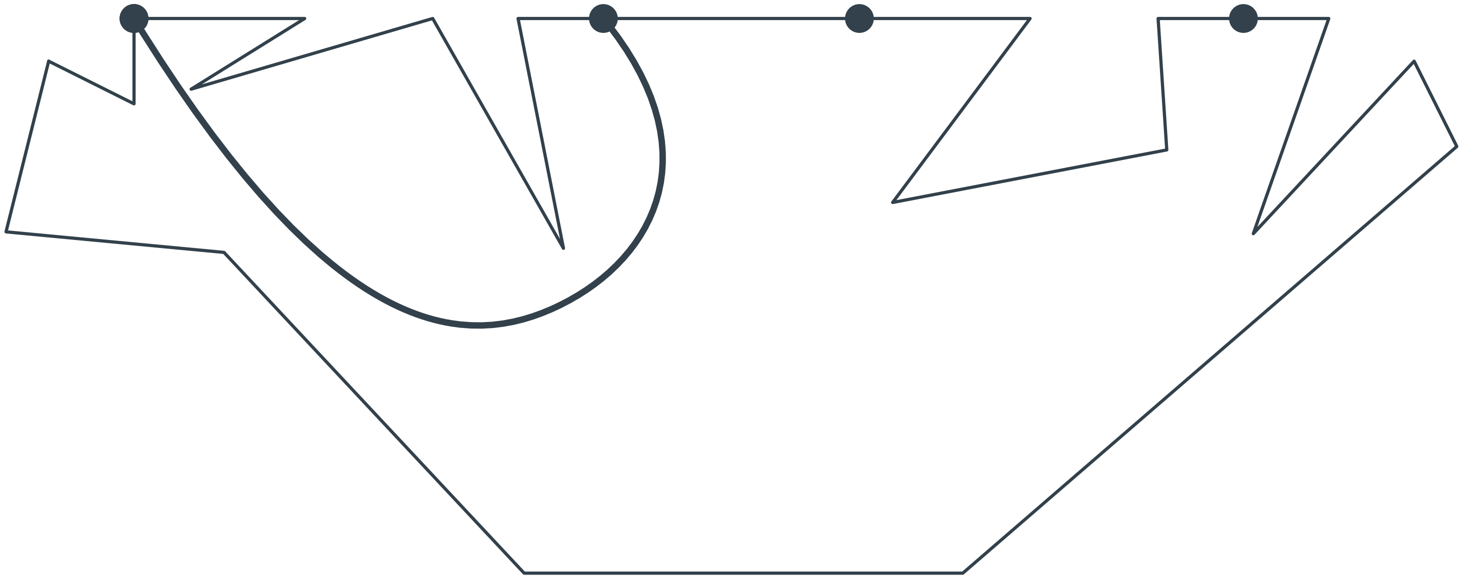
Why this order?



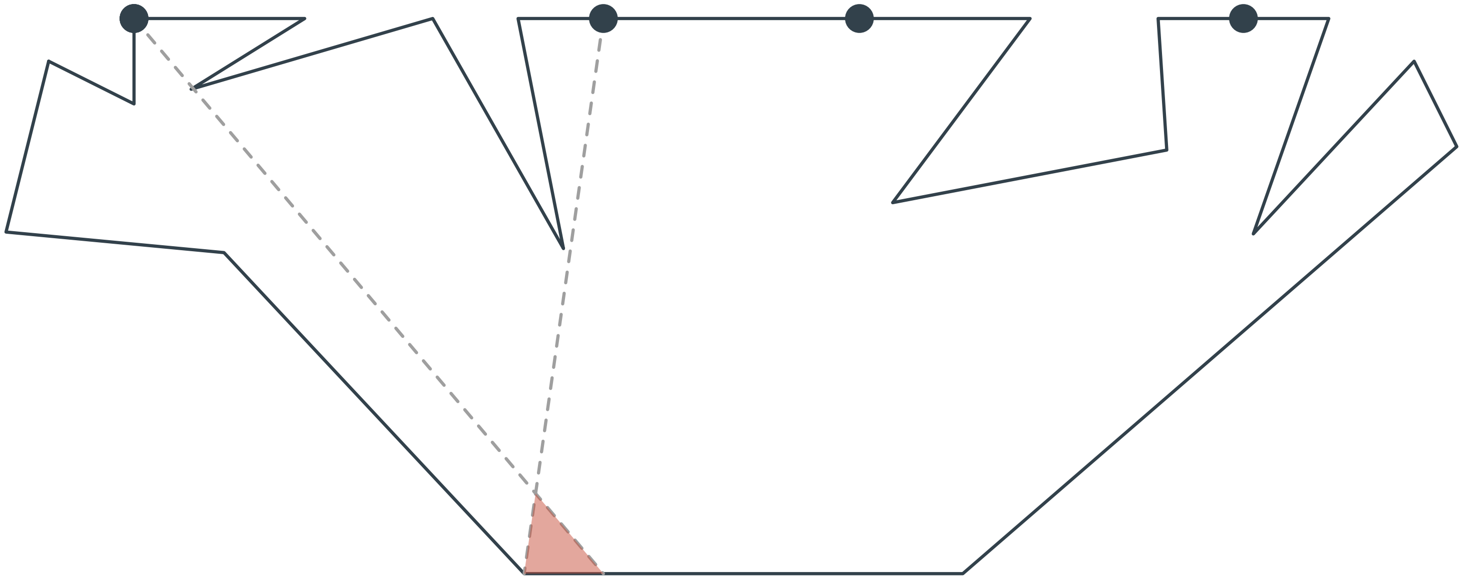
Why this order?



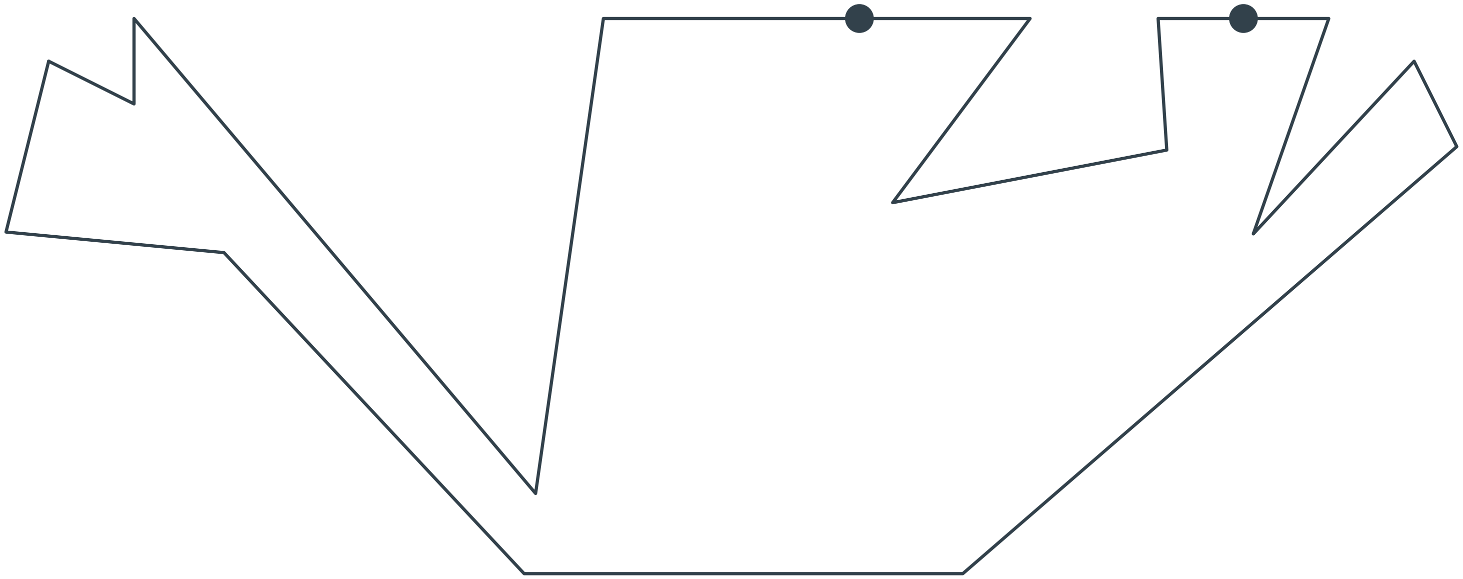
Why this order?



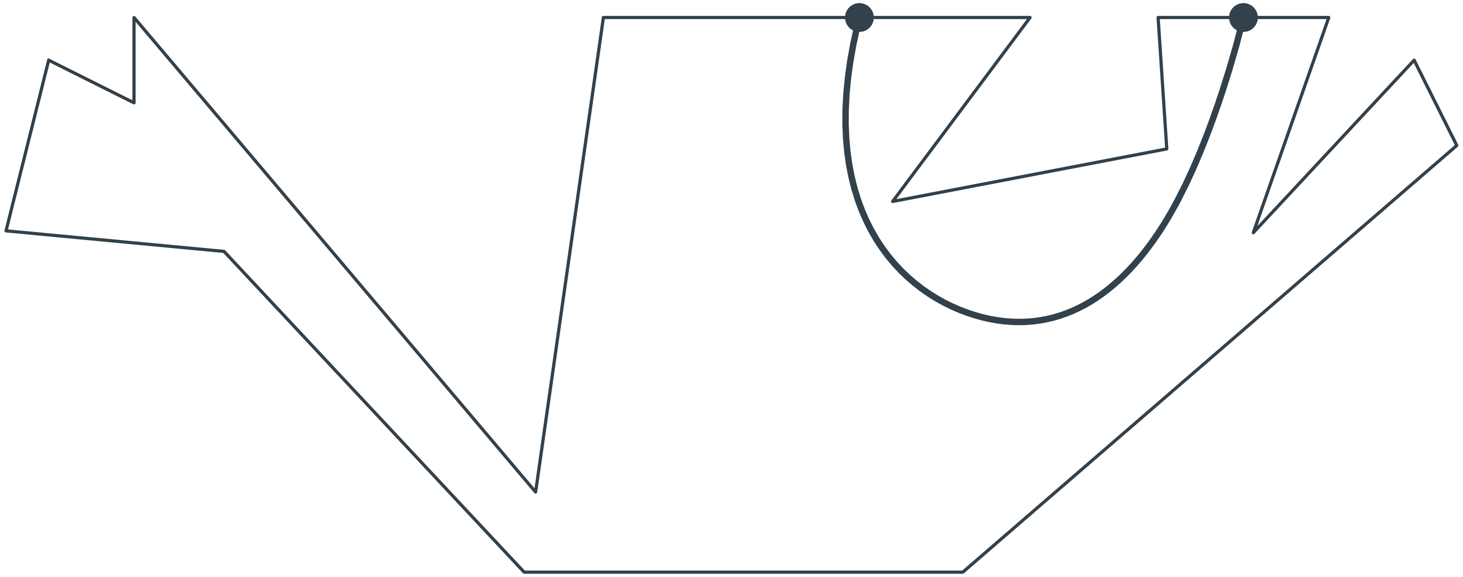
Why this order?



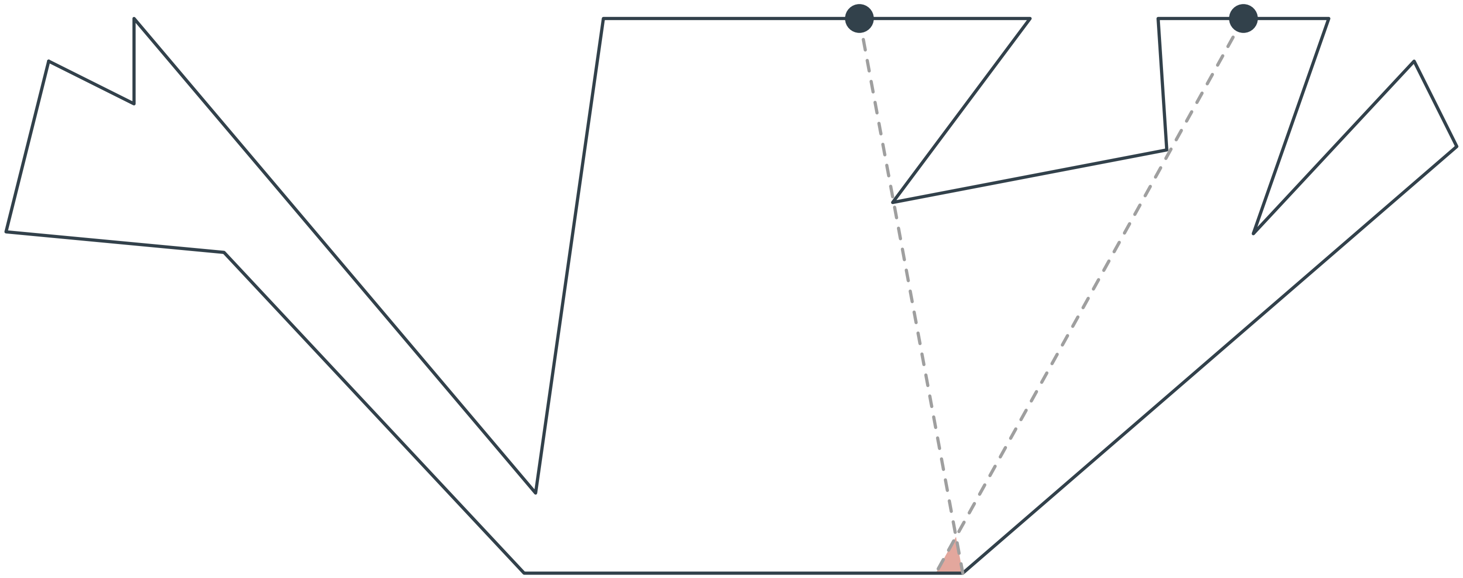
Why this order?



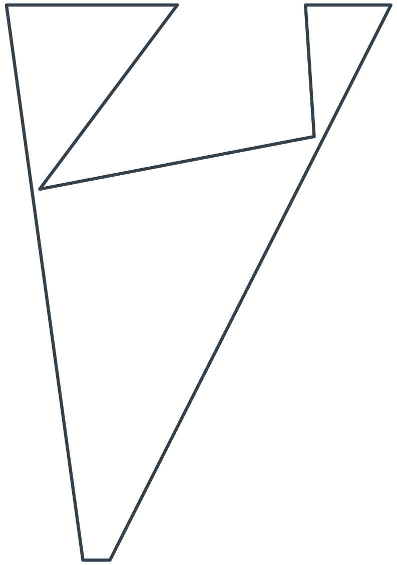
Why this order?



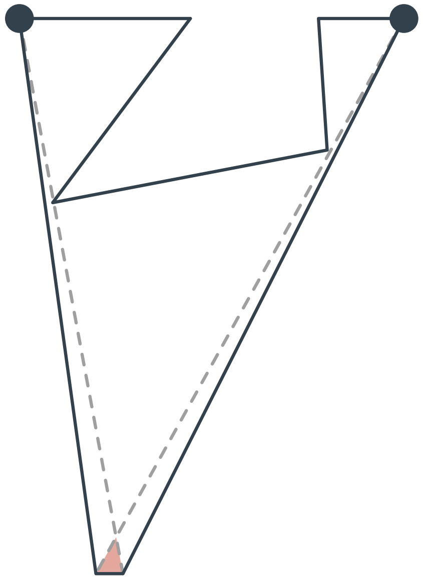
Why this order?



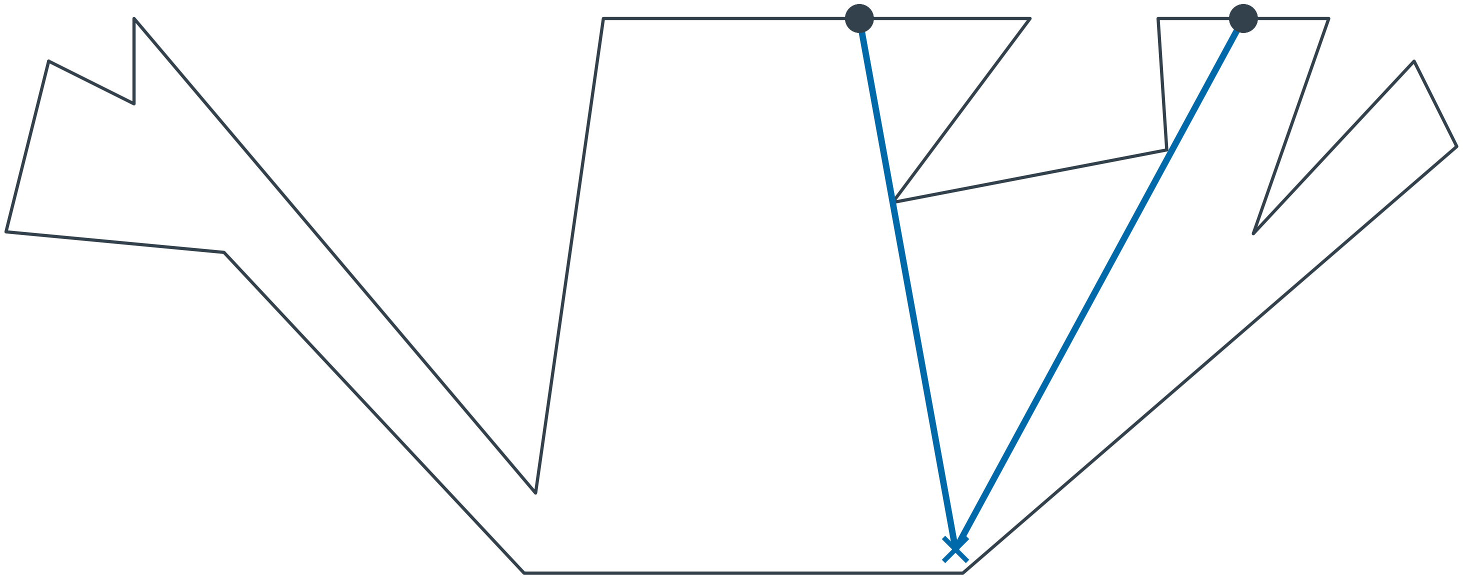
Why this order?



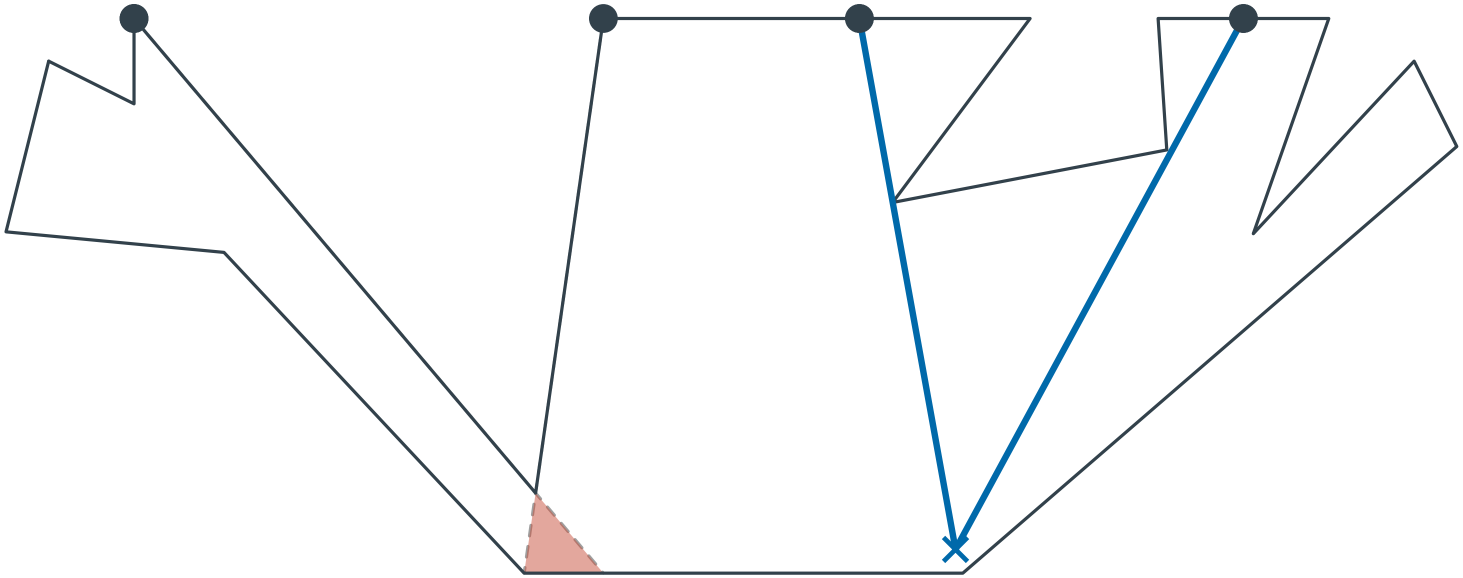
Why this order?



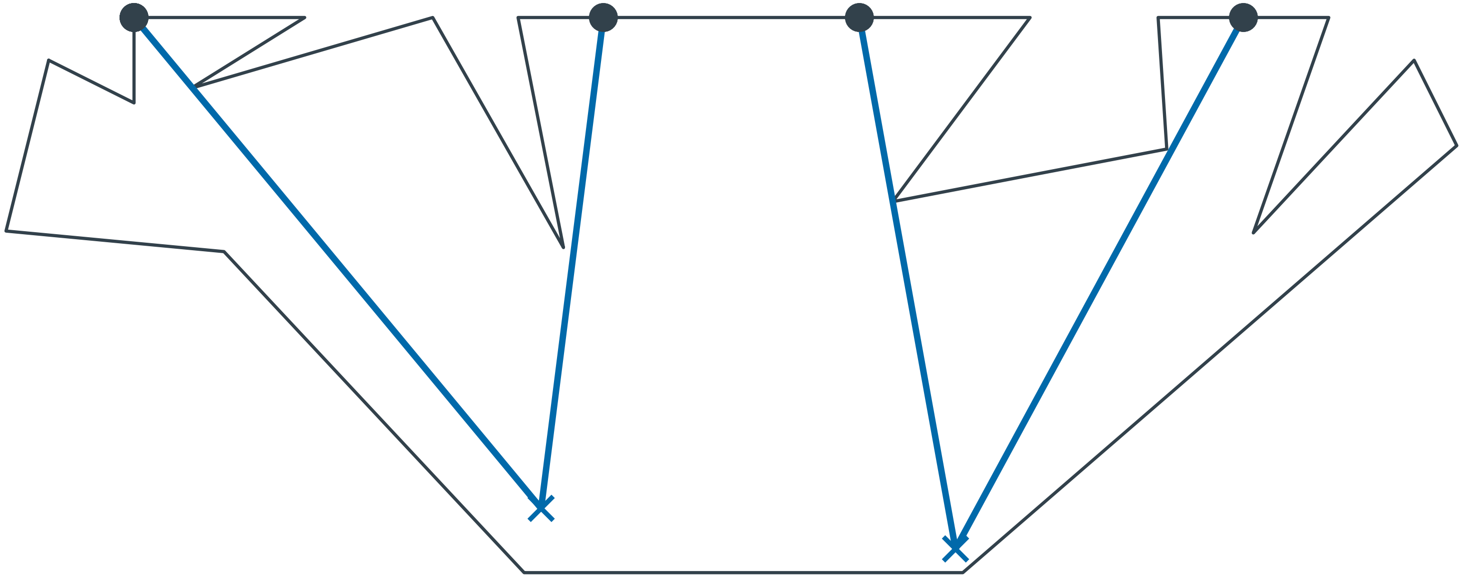
Why this order?



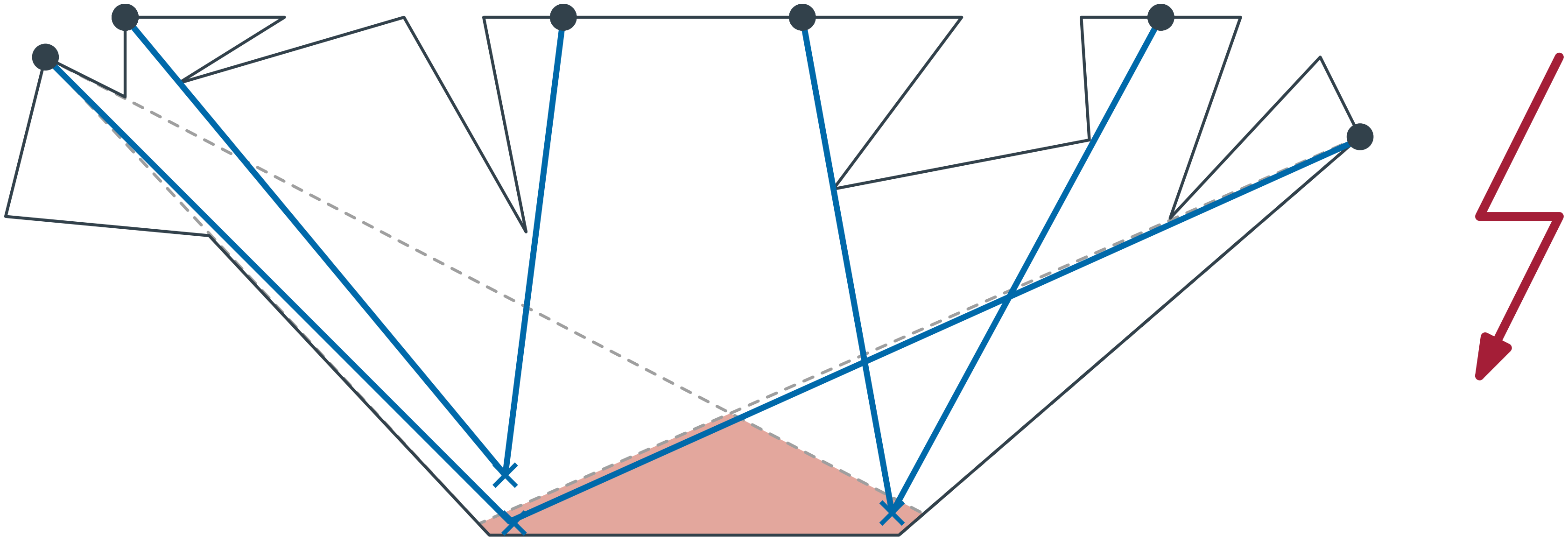
Why this order?



Why this order?



Why this order?



Running time

n:# vertices
m: # interior edges

- Tree has $O(m)$ nodes
- Per node: computer visibility polygon $O(n)$ (each refined polygon has $O(n)$ edges)

$\Longrightarrow$ $O(mn)$ time to decide whether G admits an outerplanar drawing with at ≤ 1 bend per edge that contains the given drawing of the outer face

In the positive case, the drawing can also be computed

Open problems

- More bends?
- Outer 1-planar drawings?