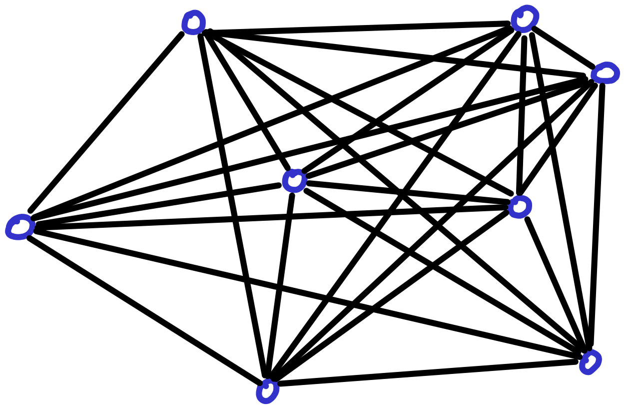
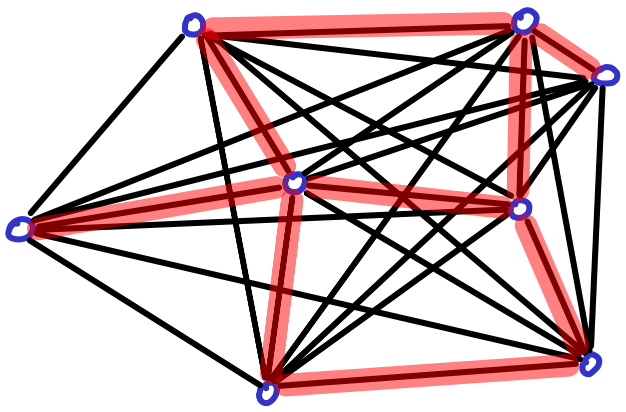
$K_n = (V, E)$: a network allowing efficient communication

... but it is expensive : $\binom{n}{2}$ edges



$K_n = (V, E)$: a network allowing efficient communication

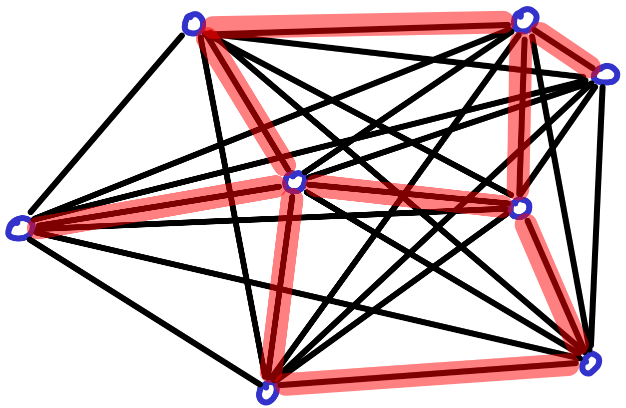
... but it is expensive : $\binom{n}{2}$ edges



Can we use a (less costly) subset $G = (V, E')$ that still ensures reasonable communication?

$K_n = (V, E)$: a network allowing efficient communication

... but it is expensive : $\binom{n}{2}$ edges



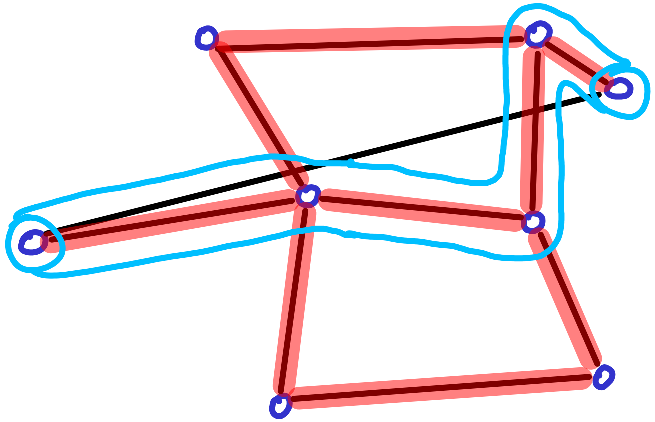
Can we use a (less costly) subset $G = (V, E')$ that still ensures reasonable communication?

Measuring

- cost → # edges
- "reasonable" → (max) detour

$K_n = (V, E)$: a network allowing efficient communication

... but it is expensive : $\binom{n}{2}$ edges



Can we use a (less costly) subset $G = (V, E')$

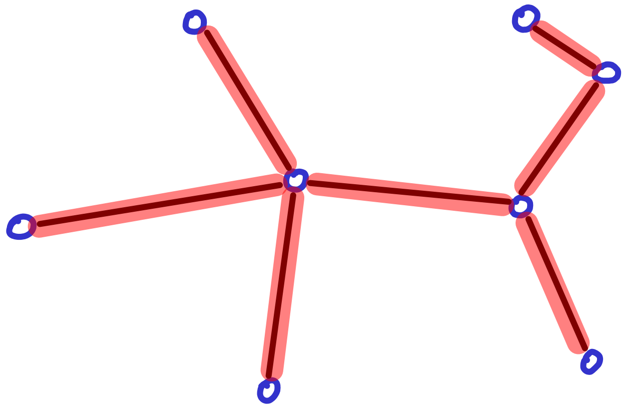
that still ensures reasonable communication?

Measuring

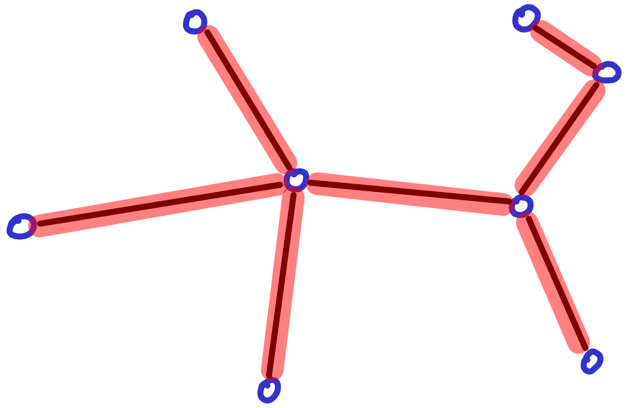
- cost → # edges
- "reasonable" → (max) detour → $\dfrac{d_G(a,b)}{d_{K_n}(a,b)}$ → shortest path vs Euclidean

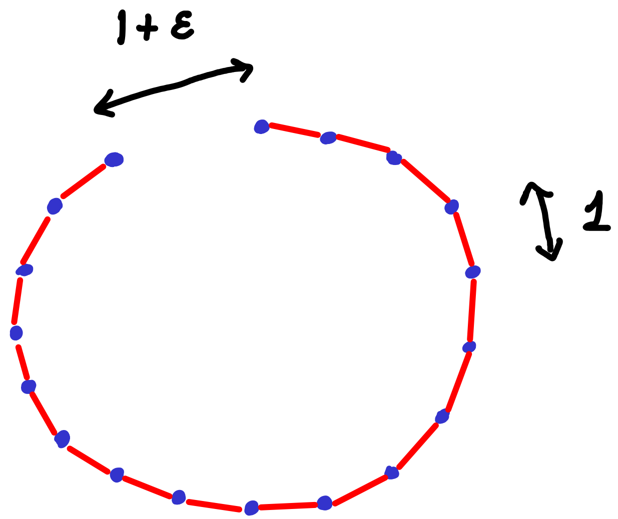
over all pairs of vertices a,b

Does the MST give a good detour ratio?



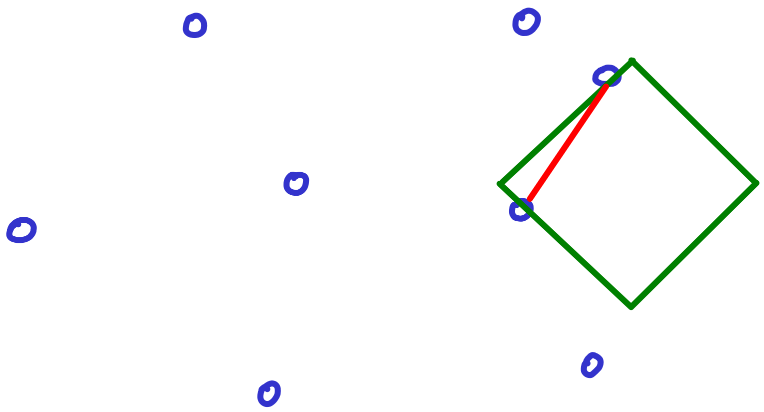
Does the MST give a good detour ratio? **NO**





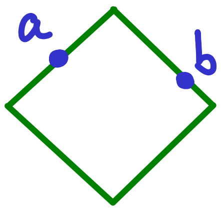
detour $= n$

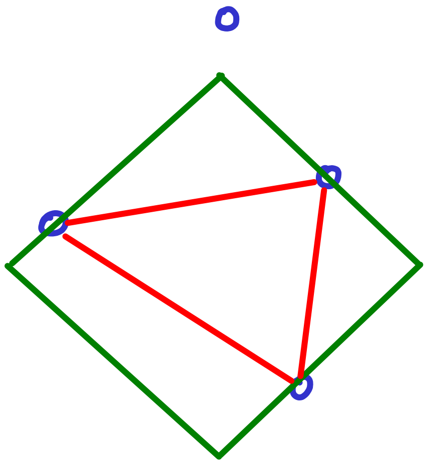
Graph T_{L_1} : keep any edge $\overline{a,b}$ iff

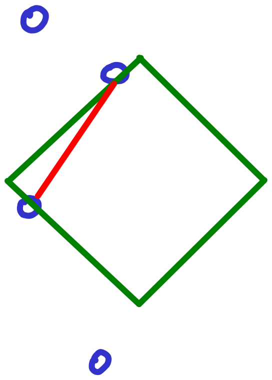


a,b are on some empty "diamond"

↳ square tilted 45°

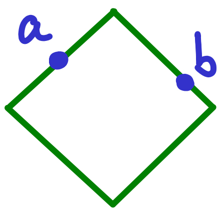






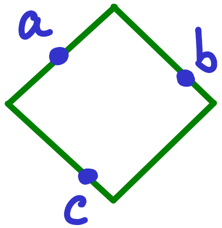
T_{L_1} : keep any edge $\overline{a,b}$ iff

a,b are on some empty "diamond"

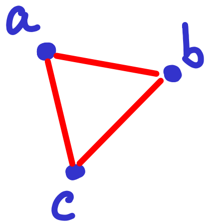


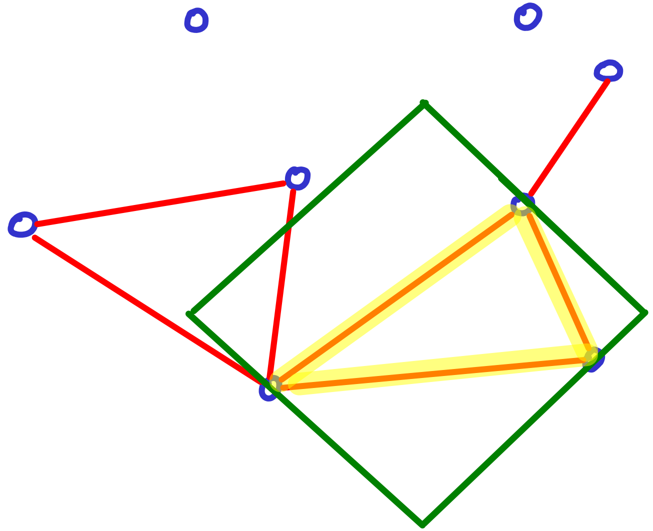
↳ square tilted 45°

Notice if



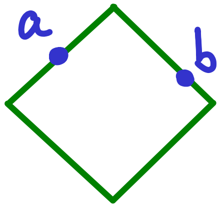
then





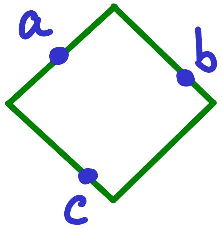
T_{L_1} : keep any edge $\overline{a,b}$ iff

a,b are on some empty "diamond"

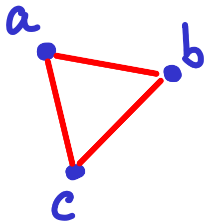


↳ square tilted 45°

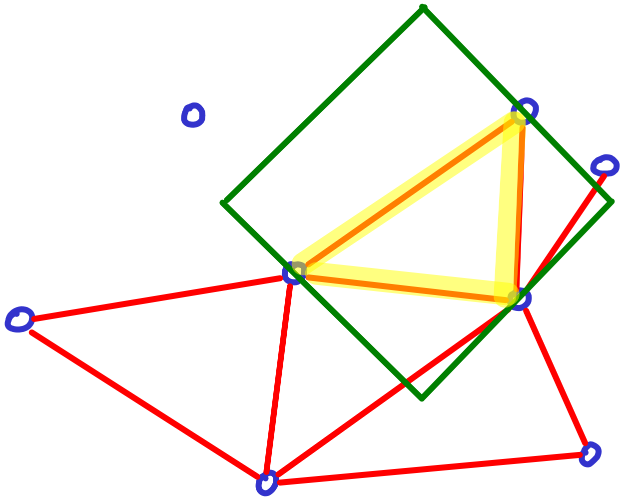
Notice if



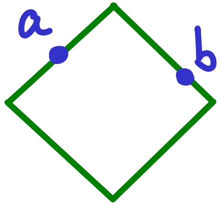
then



T_{L_1} : keep any edge $\overline{a,b}$ iff

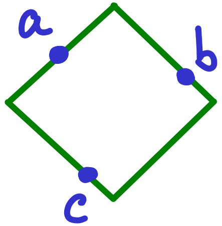


a,b are on some empty "diamond"

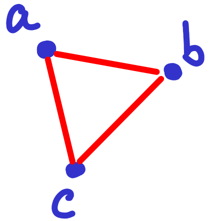


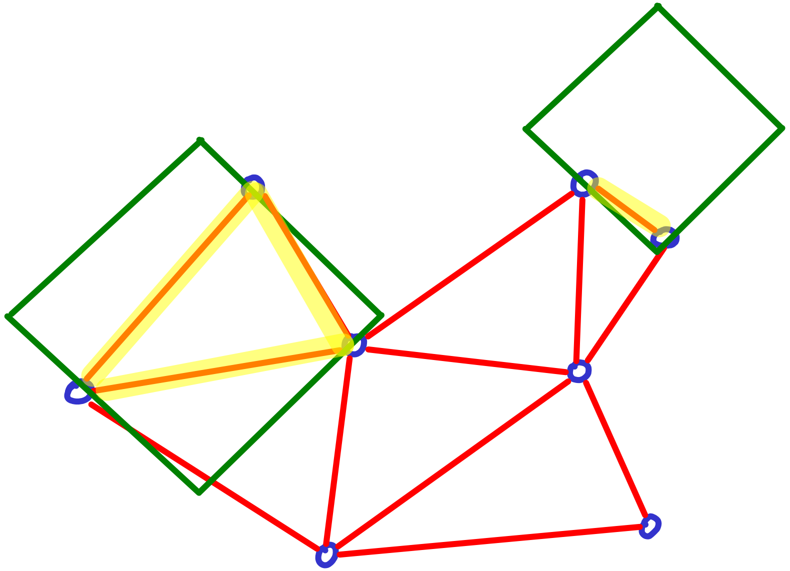
↳ square tilted 45°

Notice if

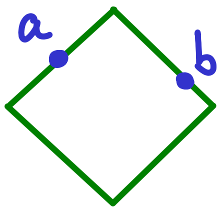


then

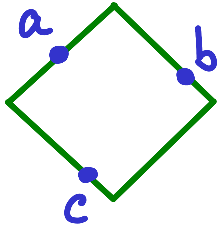


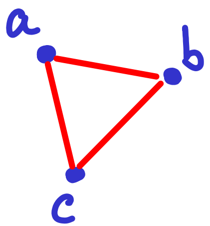


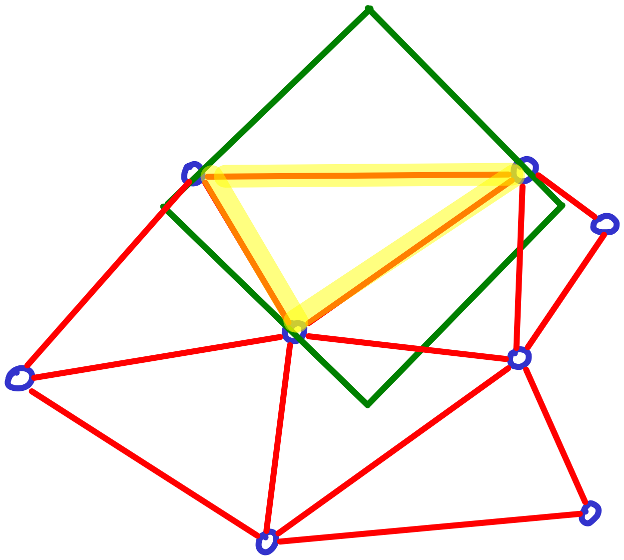
T_{L_1} : keep any edge $\overline{a,b}$ iff a,b are on some empty "diamond"



↳ square tilted 45°

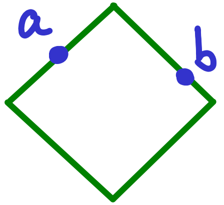
Notice if 

then 



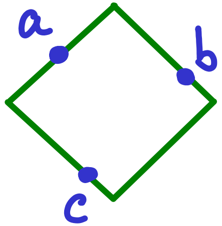
T_{L_1} : keep any edge $\overline{a,b}$ iff

a,b are on some empty "diamond"

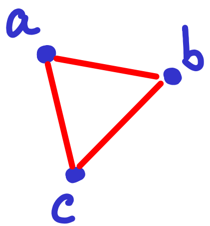


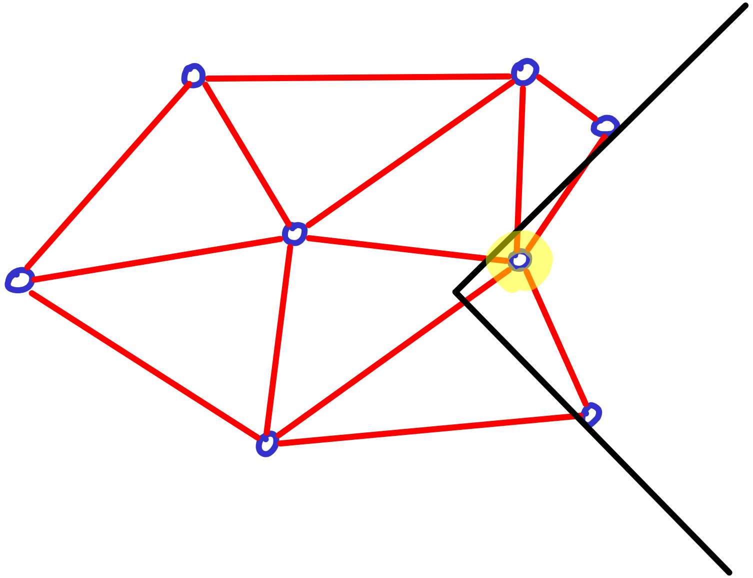
↳ square tilted 45°

Notice if



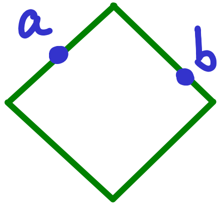
then





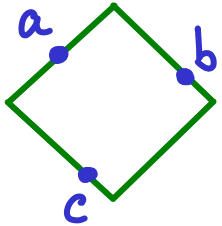
T_{L_1} : keep any edge $\overline{a,b}$ iff

a,b are on some empty "diamond"

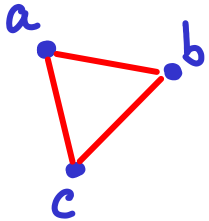


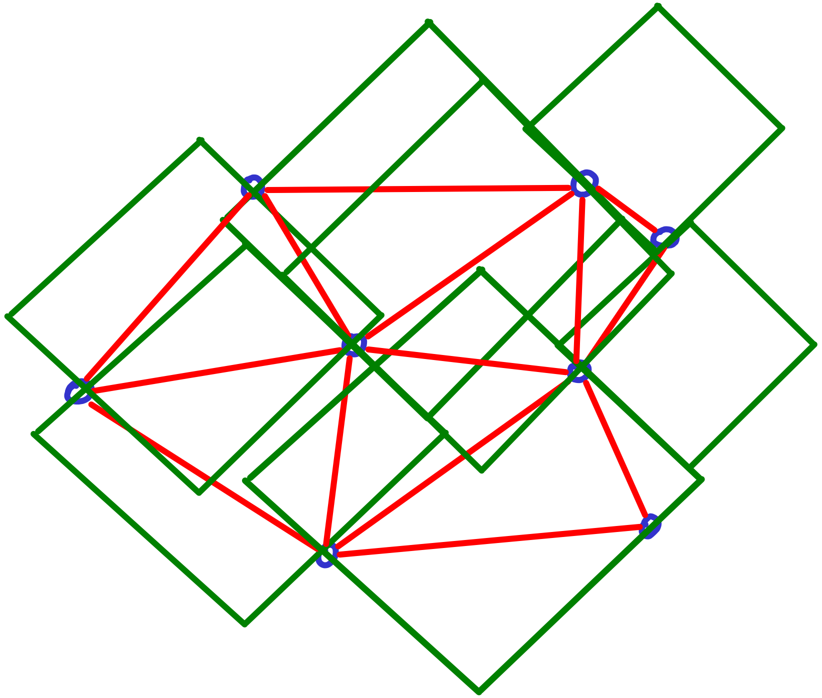
↳ square tilted 45°

Notice if



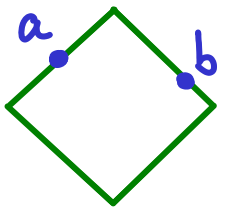
then





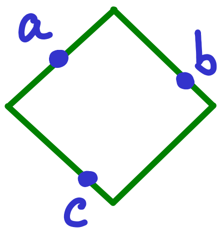
T_{L_1} : keep any edge $\overline{a,b}$ iff

a,b are on some empty "diamond"

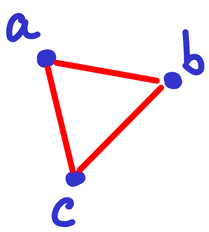


↳ square tilted 45°

Notice if



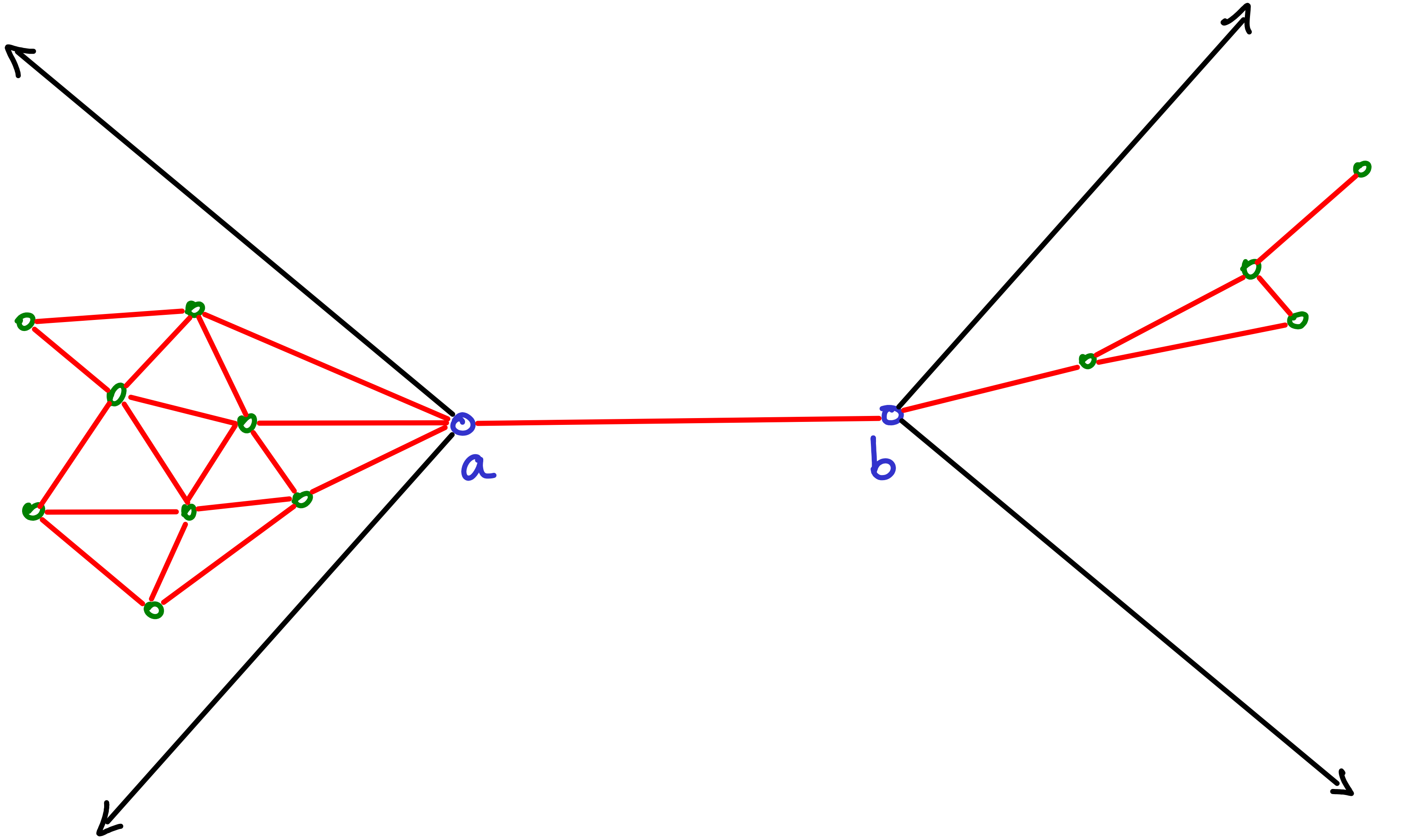
then



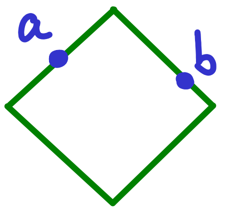
Why T_{L_1} ?

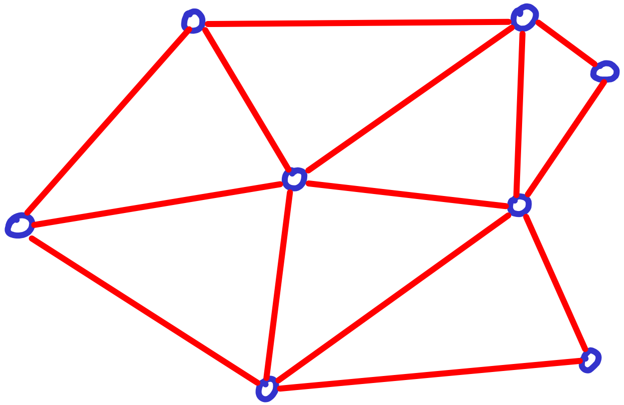
T is for triangular faces

Why T_{L_1} ?



T_{L_1} : keep any edge $\overline{a,b}$ iff a, b are on some empty "diamond"



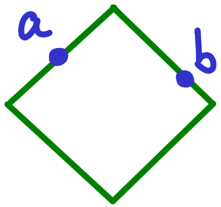


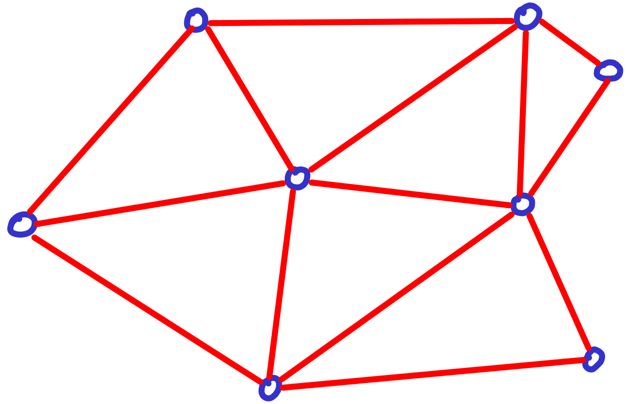
Assume "general position"

↳ no 4 points on an empty diamond

(it is just a technicality)

T_{L_1} : keep any edge $\overline{a,b}$ iff a,b are on some empty "diamond"





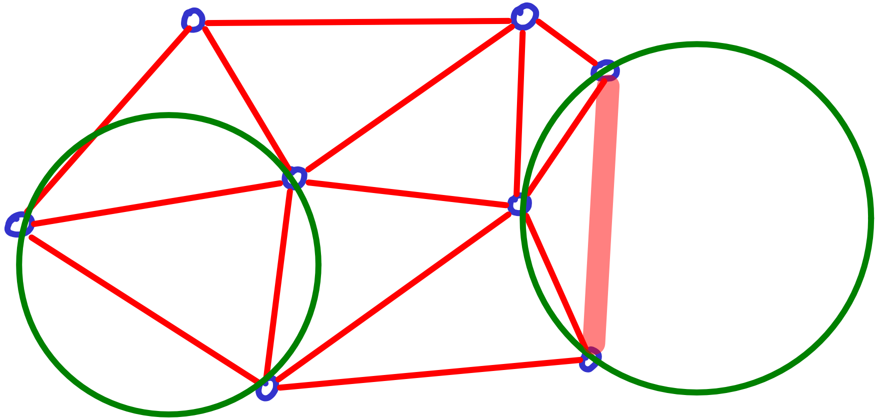
Assume "general position"

↳ no 4 points on an empty diamond

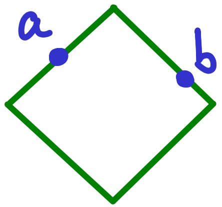
(it is just a technicality)

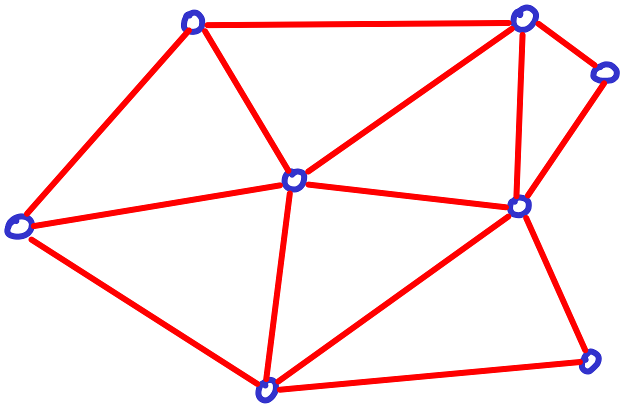
T_{L_2} : Delaunay Triangulation

based on empty circles



T_{L_1} : keep any edge $\overline{a,b}$ iff a,b are on some empty "diamond"





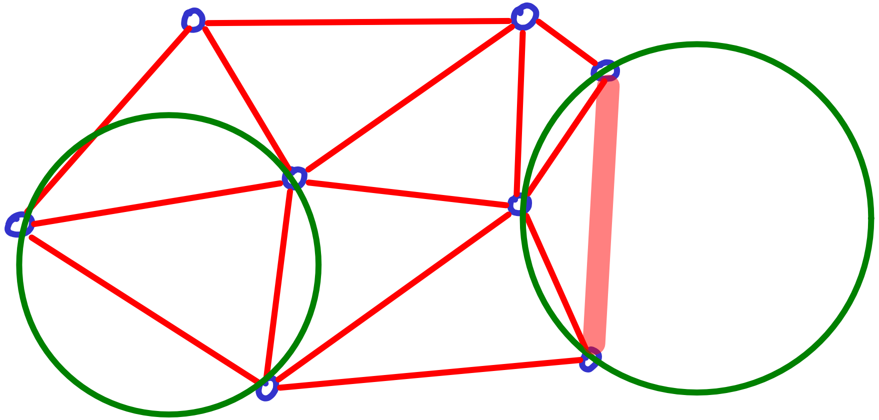
Assume "general position"

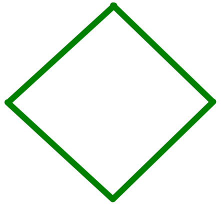
↳ no 4 points on an empty diamond

(it is just a technicality)

T_{L_2} : Delaunay Triangulation

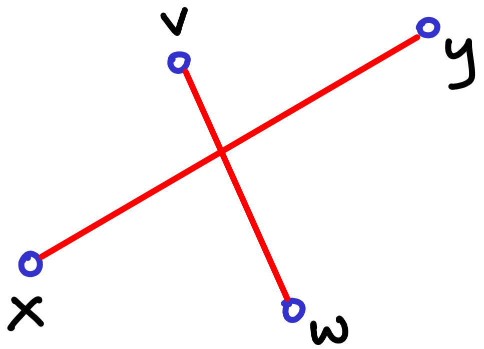
based on empty circles

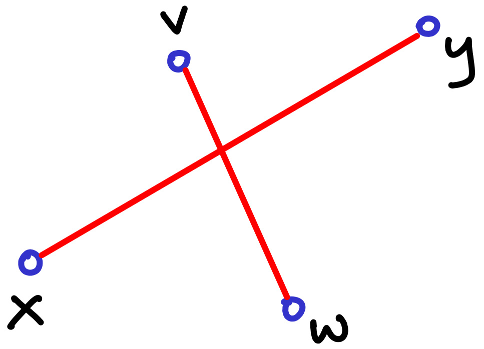




is a circle in the L_1 metric

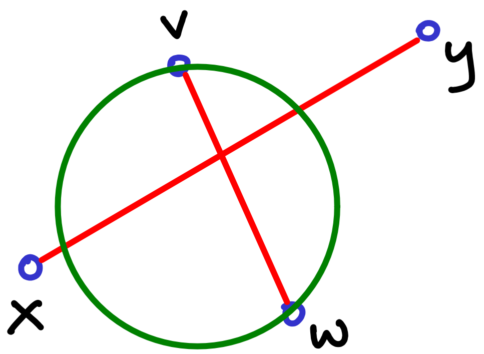
Can 2 edges cross in T_{L_2} ?





Can 2 edges cross in T_{L_2} ?

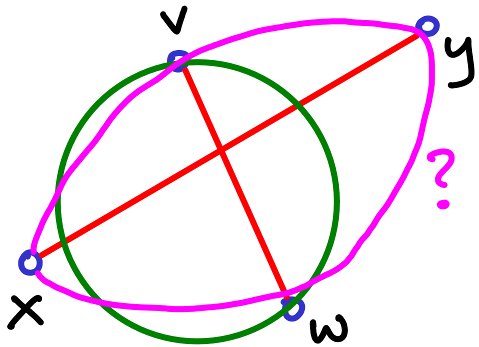
Suppose $\overline{xy}$ crosses $\overline{vw}$



Can 2 edges cross in T_{L_2} ?

Suppose $\overline{xy}$ crosses $\overline{vw}$

$\exists$ empty circle ○ through v,w

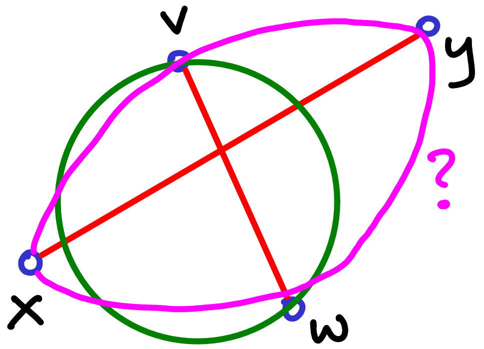


Can 2 edges cross in T_{L_2} ?

Suppose $\overline{xy}$ crosses $\overline{vw}$

$\exists$ empty circle ○ through v, w

$\exists$ " " ○ " x, y.



Can 2 edges cross in T_{L_2} ?

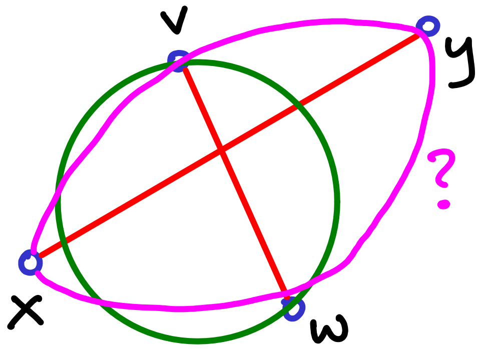
Suppose $\overline{xy}$ crosses $\overline{vw}$

∃ empty circle ◯ through v,w

∃ " " ◯ " x,y.

Assuming general position (no 4 on a circle)

◯ ◯ are distinct & must intersect exacty twice.



Can 2 edges cross in T_{L_2}? <u>No</u>

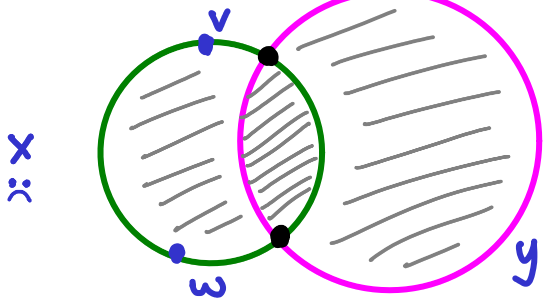
Suppose $\overline{xy}$ crosses $\overline{vw}$

∃ empty circle ○ through v,w

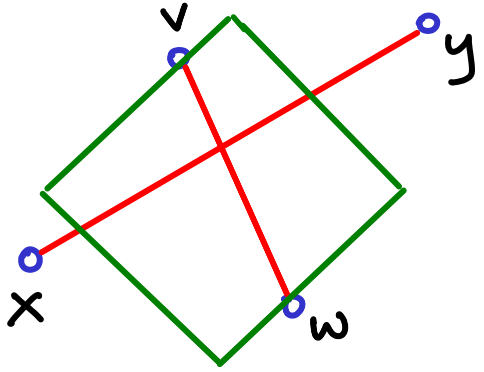
∃ " " ○ " x,y.

Assuming general position (no 4 on a circle)

○ ○ are distinct & must intersect exacty twice.



No way to place x,y,v,w.



Can 2 edges cross in T_{L_1} ? <u>NO</u>

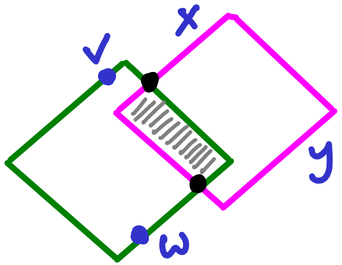
Suppose $\overline{xy}$ crosses $\overline{vw}$

∃ empty "circle" ◇ through v,w

∃ " " ◇ " x,y.

Assuming general position (no 4 on a ◇)

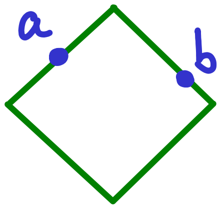
◇ ◇ are distinct & must intersect exacty twice.

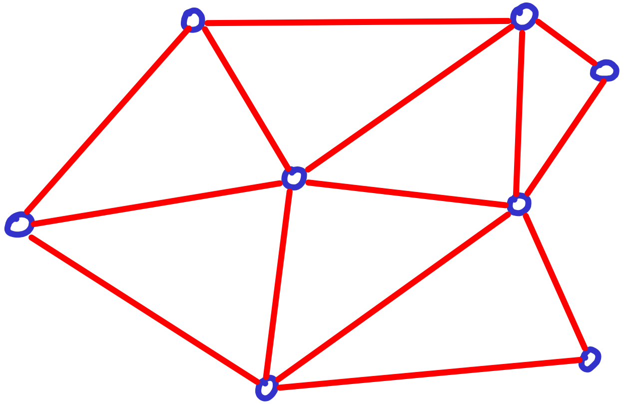


No way to place x,y,v,w.

technical details omitted

T_{L_1} : keep any edge $\overline{a,b}$ iff a,b are on some empty diamond

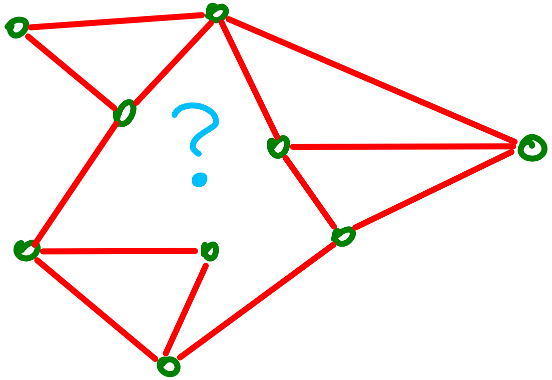




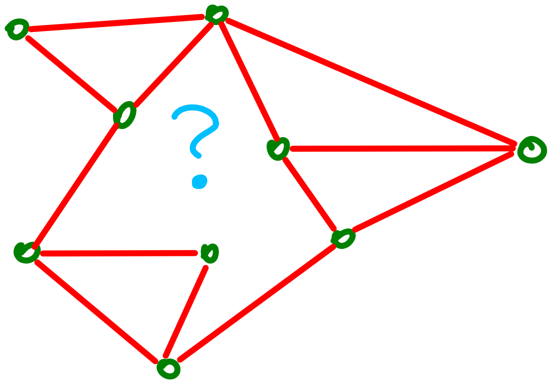
No edges cross : planar graph

↳ $O(V)$ edges by Euler

Can a bounded face not be a triangle?

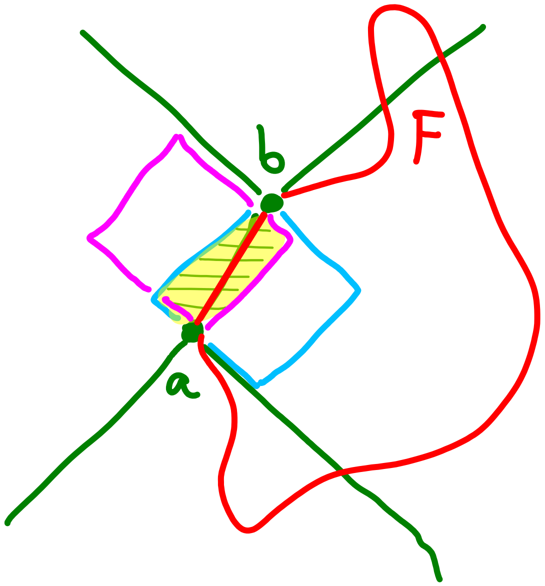


Can a bounded face not be a triangle?

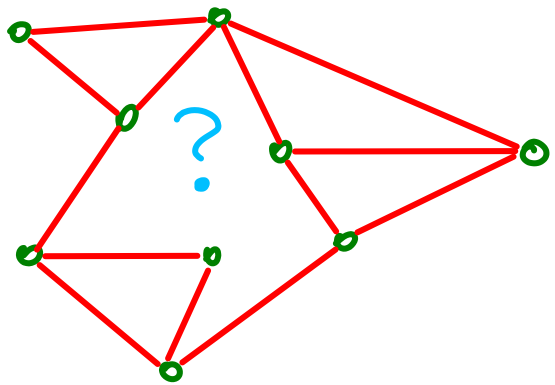


Let $\overline{ab} \in F \neq$ triangle

They define an empty region



Can a bounded face not be a triangle?

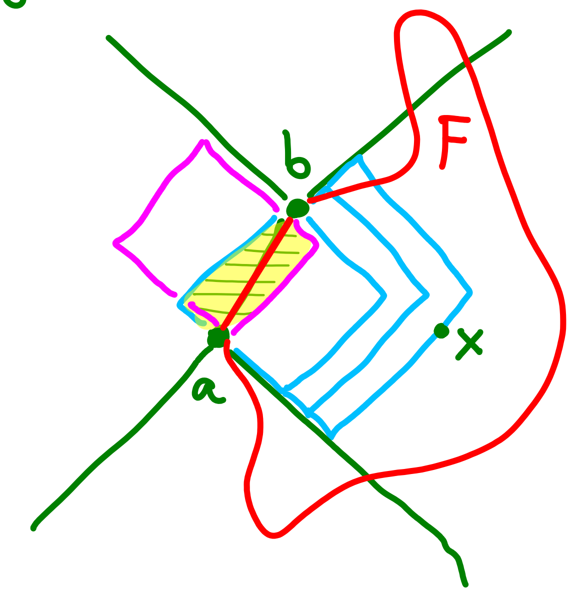


Let $\overline{ab} \in F \neq$ triangle

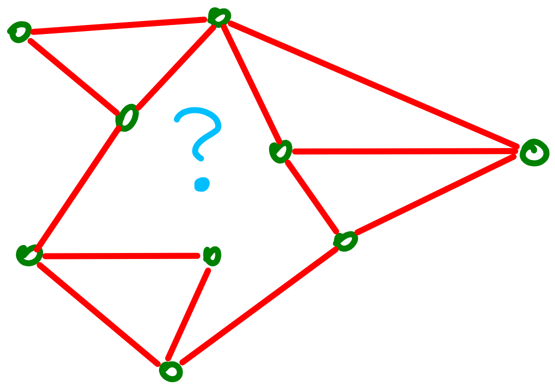
They define an empty region

from which we can grow an empty diamond "within" F.

We must hit some point $\times$ eventually.



Can a bounded face not be a triangle? $\boxed{\text{NO}}$

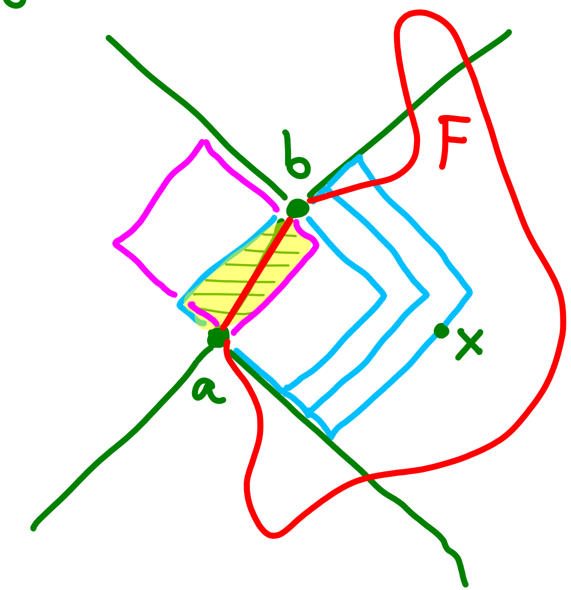


Let $\overline{ab} \in F \neq$ triangle

They define an empty region

from which we can grow an empty diamond "within" F.

We must hit some point x eventually.



If $x \in F$ we subdivide F

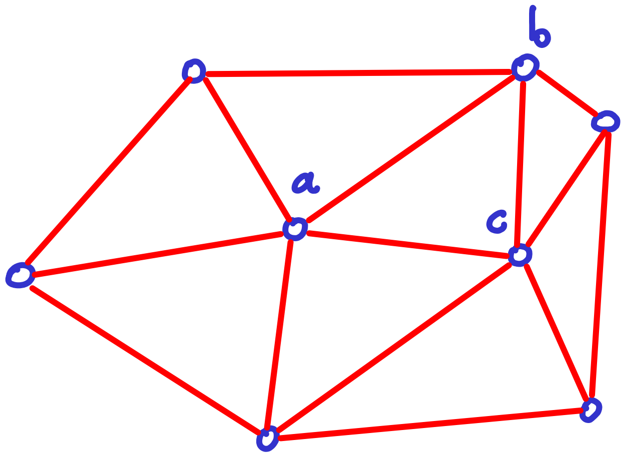
else $\overline{xa}$ & $\overline{xb}$ cross F

$\Bigg\}$ contradiction

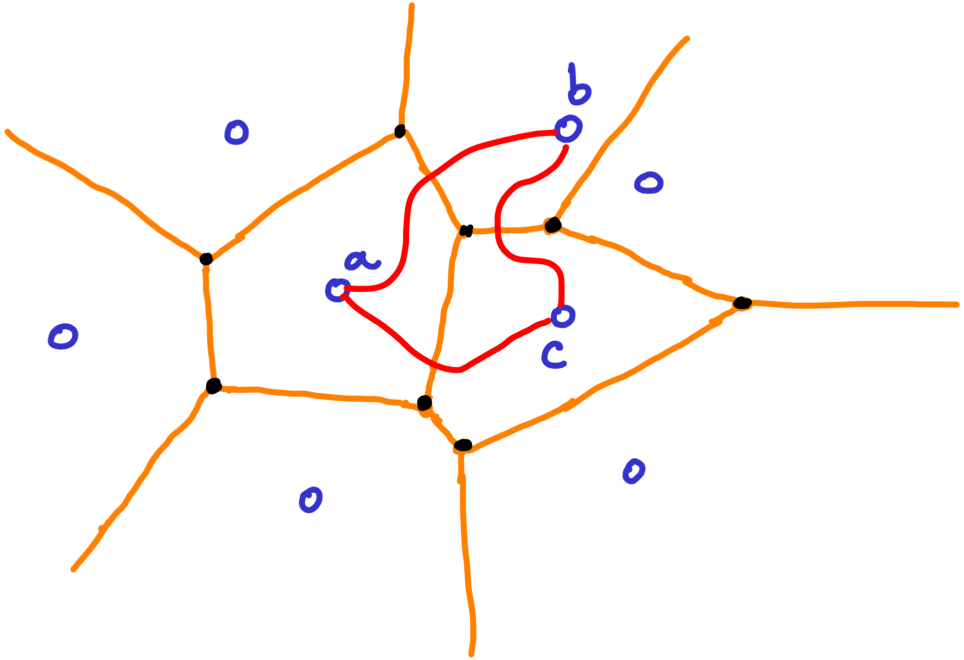
FYI

defines regions closest to input sites

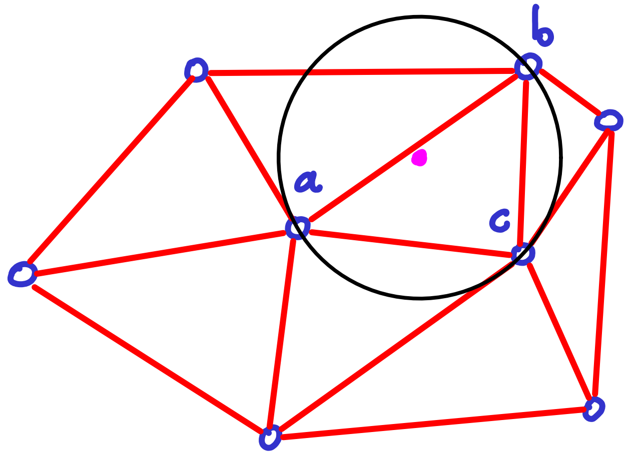
Dual of VORONOI DIAGRAM



L_2



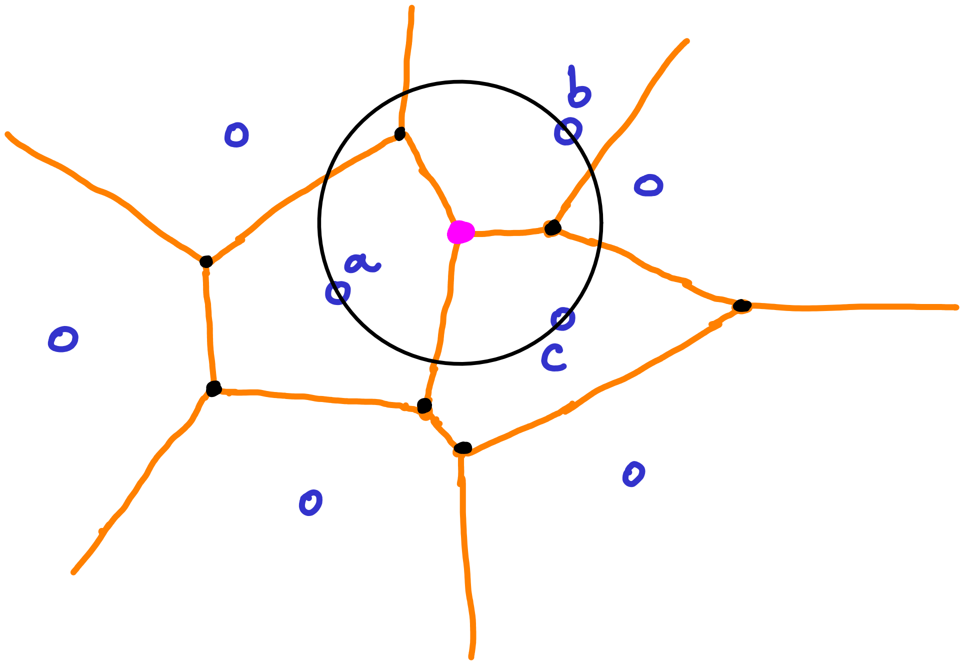
FYI



defines regions closest to input sites

Dual of <u>VORONOI DIAGRAM</u>

L_2

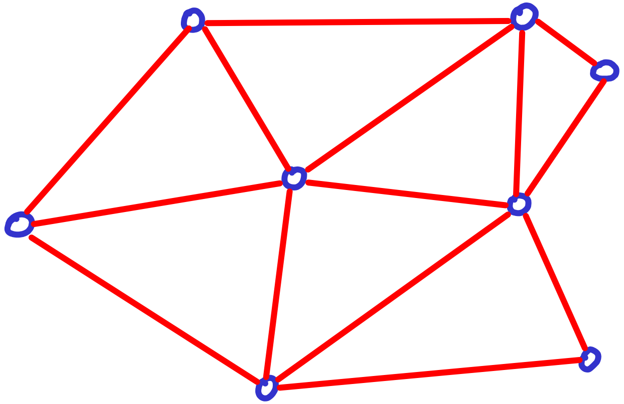


a, b, c form (empty) triangle
because ∃ empty circle on a, b, c

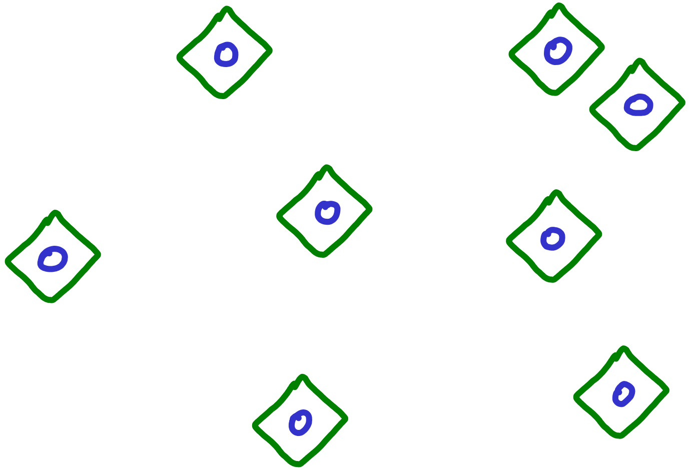
FYI

can be constructed by expanding empty "circles"

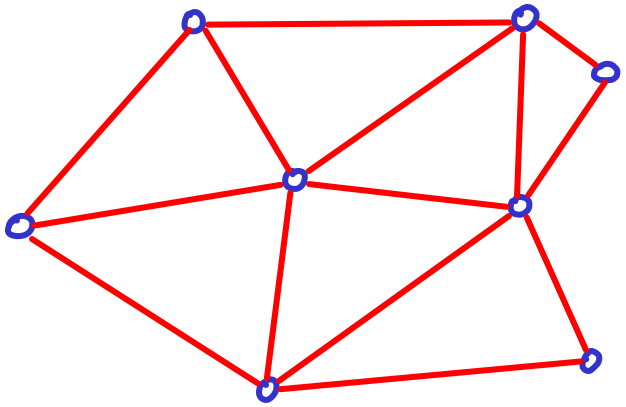
Dual of VORONOI DIAGRAM



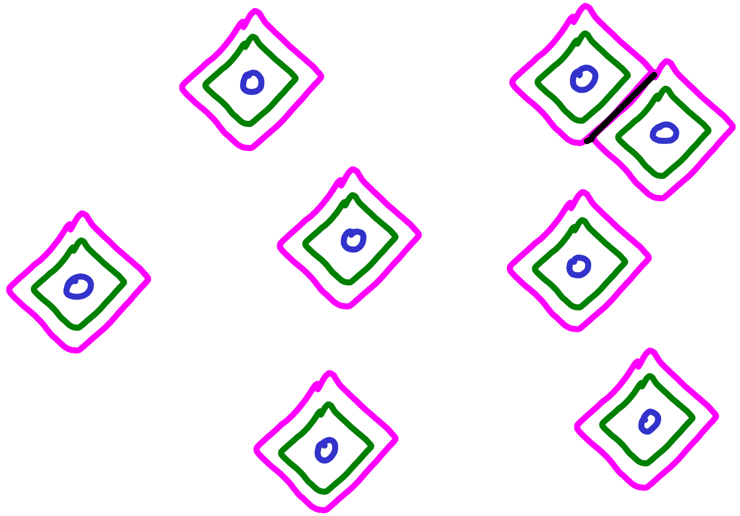
L_1



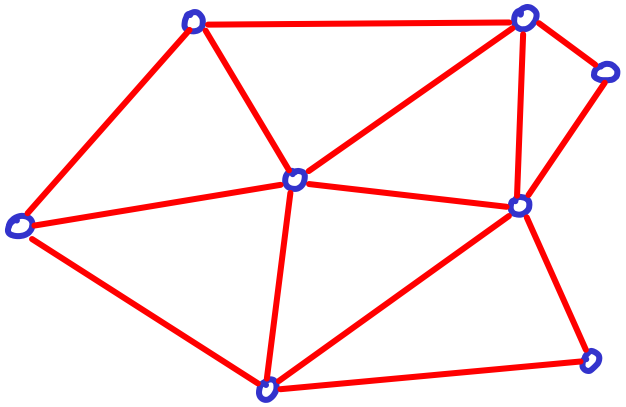
FYI



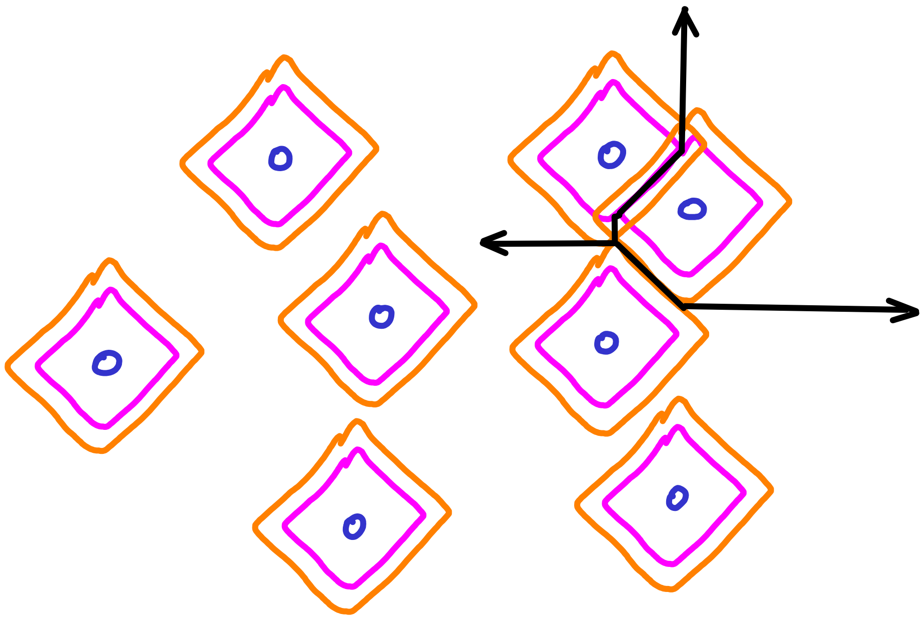
Dual of VORONOI DIAGRAM



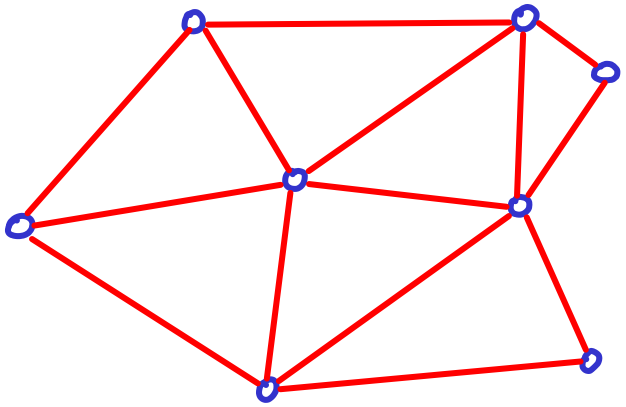
FYI



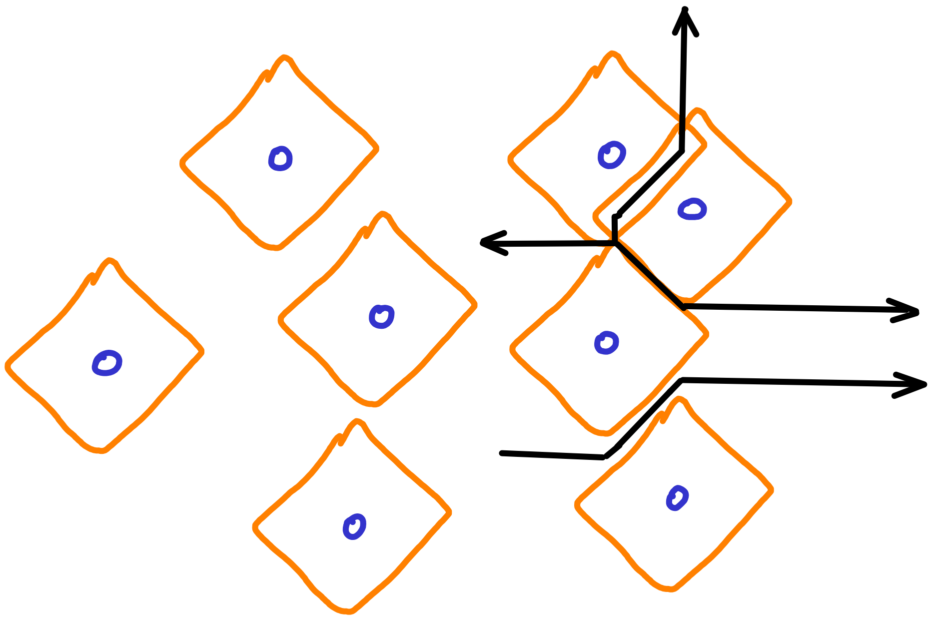
Dual of VORONOI DIAGRAM



FYI

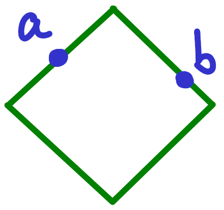


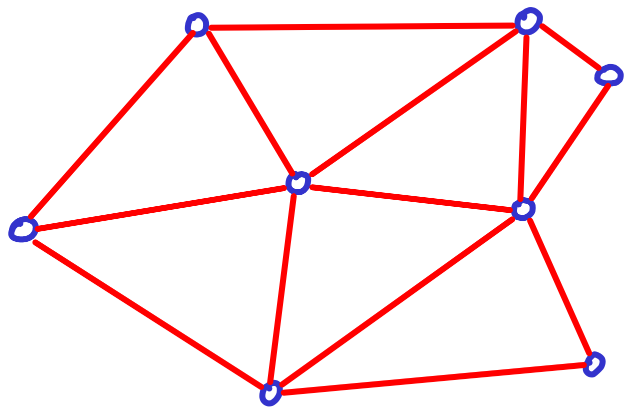
Dual of VORONOI DIAGRAM



etc

T_{L_1} : keep any edge $\overline{a,b}$ iff a,b are on some empty diamond





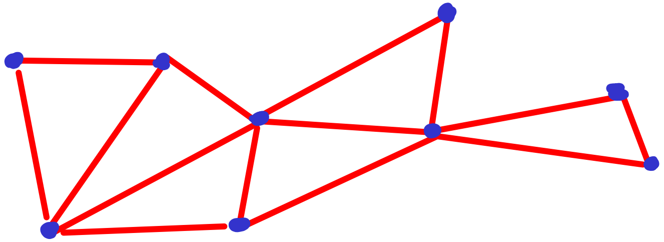
No edges cross : planar graph

↳ $O(V)$ edges.

Claim this gives a worst-case detour of $\sqrt{10}$.

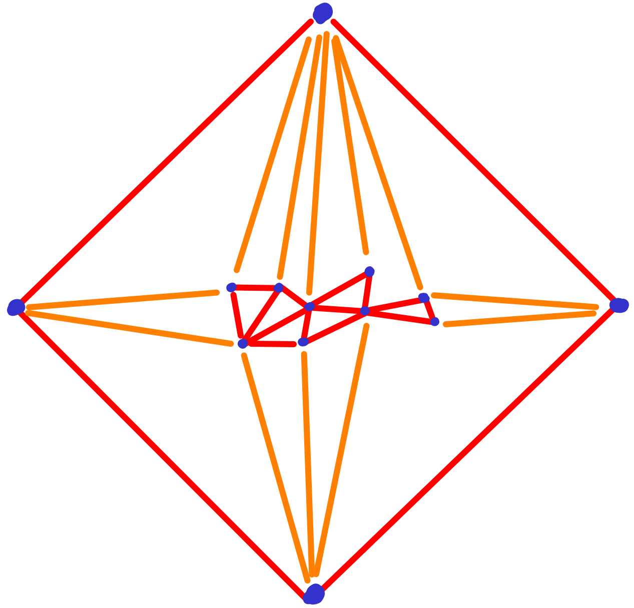
i.e. it is a "$\sqrt{10}$-spanner" (t-spanner w/ $t=\sqrt{10}$)

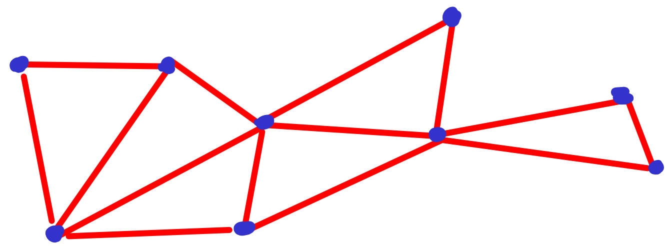
(result by Paul Chew ~1986)



to help w/ proof:

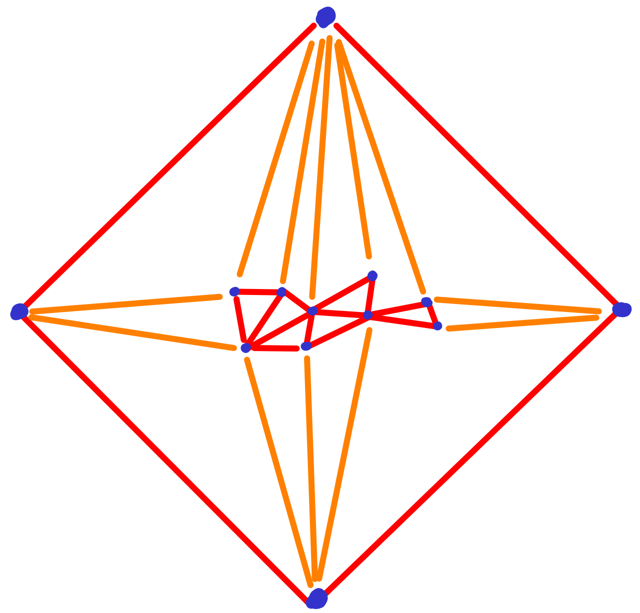
$\xrightarrow[\text{augment}]{}$





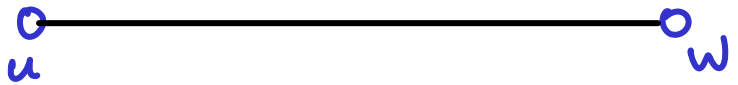
to help w/ proof:

$\xrightarrow{\text{augment}}$

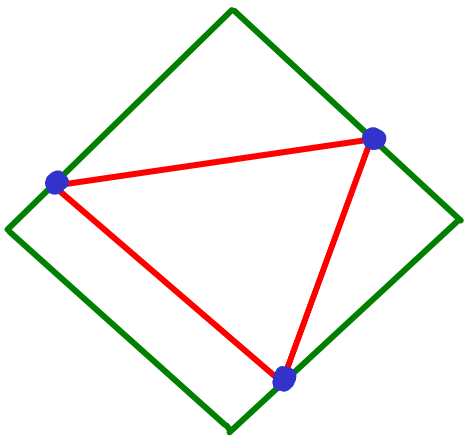


Suppose 2 points u, w have same y-coord:

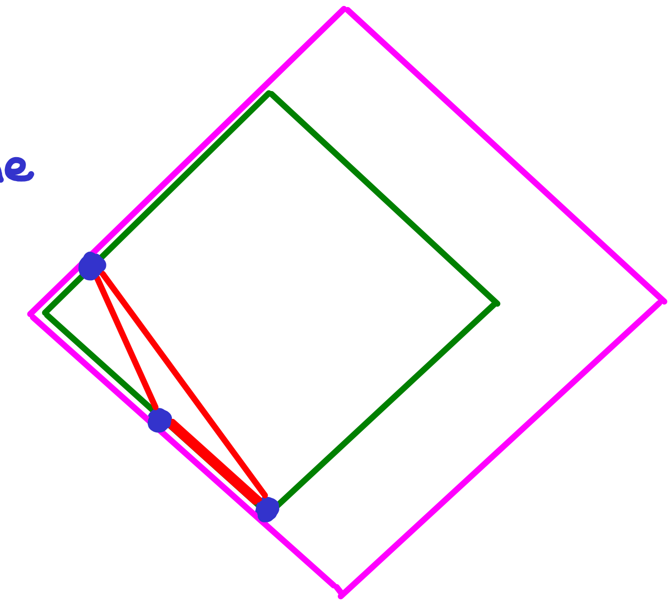
We will sketch why T_{L_1} is a $\sqrt{8}$-spanner for $\overline{uw}$.



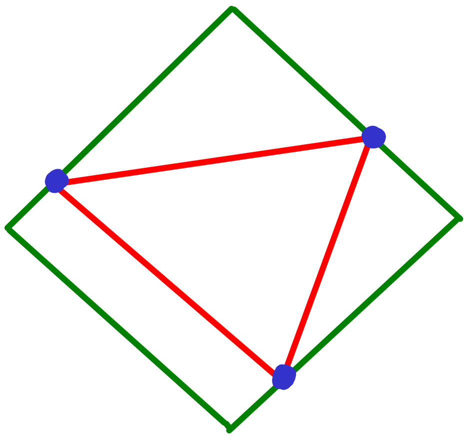
3 points either define a unique diamond



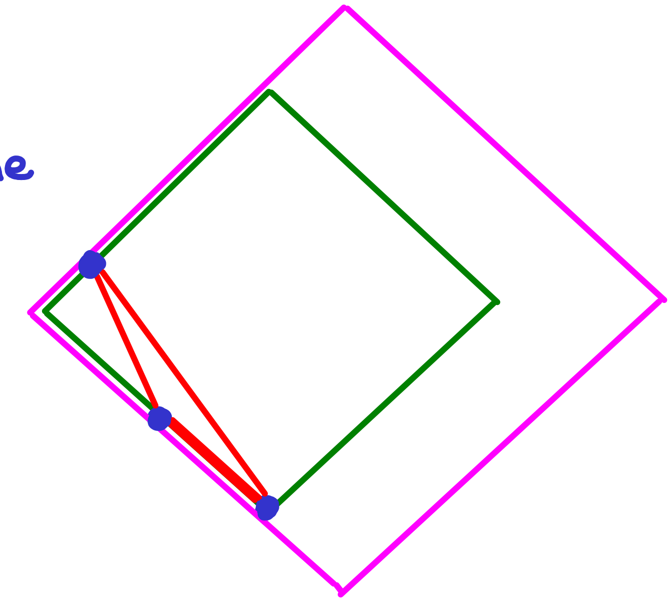
or we use a minimal one

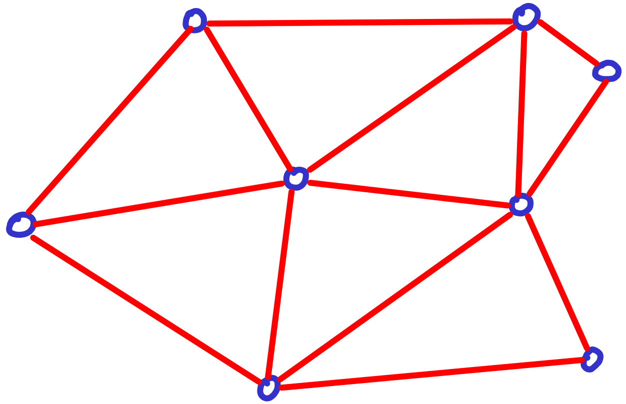


3 points either define a unique diamond

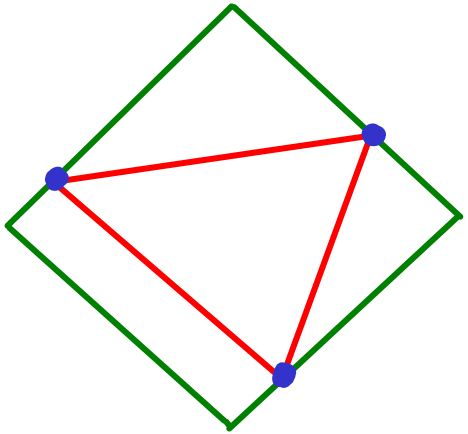


or we use a minimal one

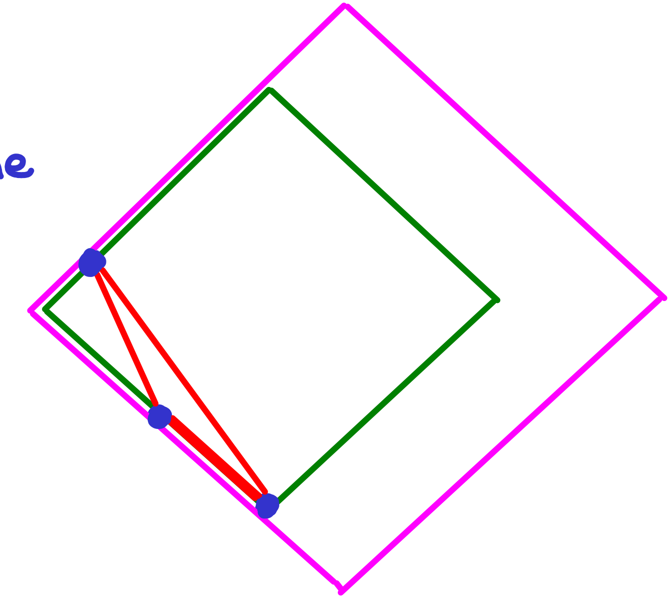


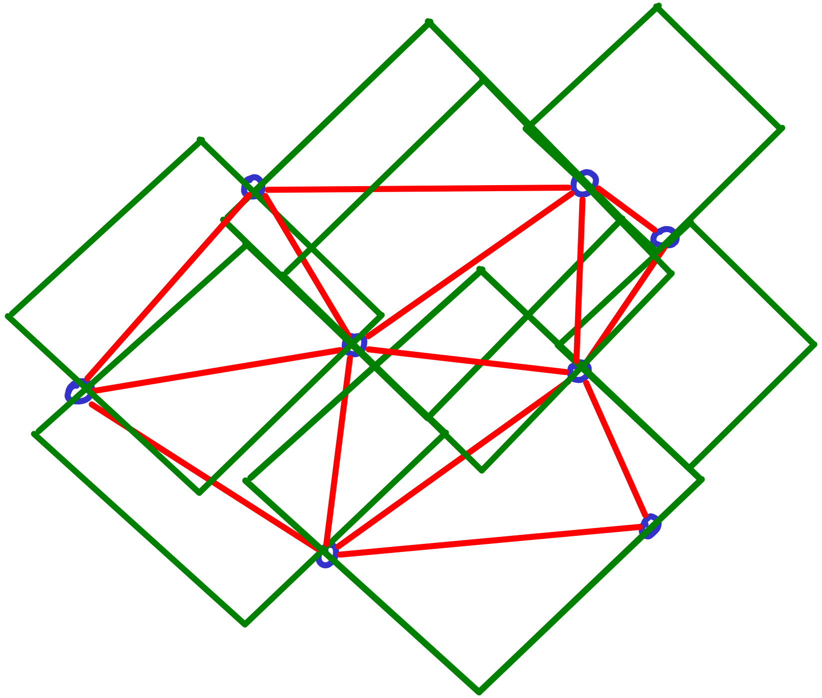


3 points either define a unique diamond

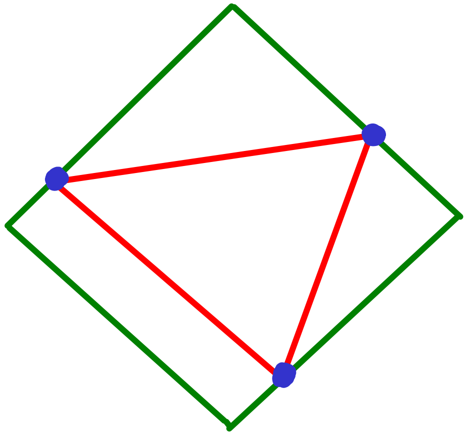


or we use a minimal one

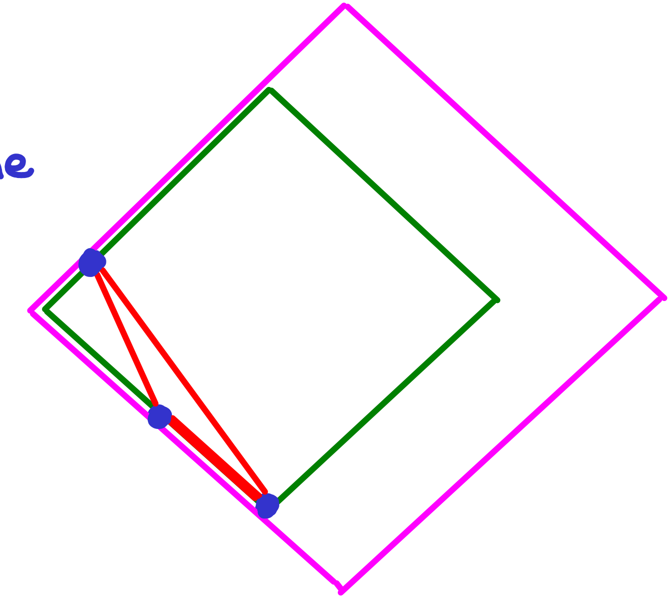


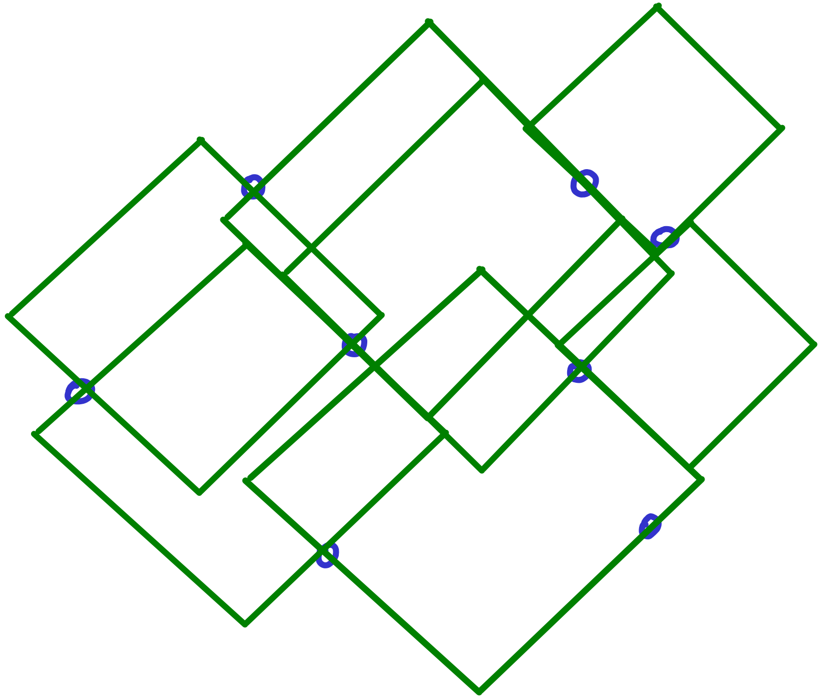


3 points either define a unique diamond

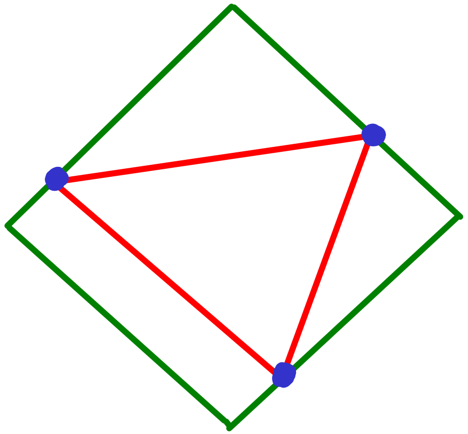


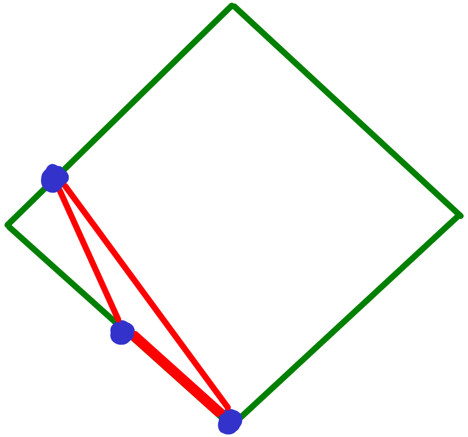
or we use a minimal one



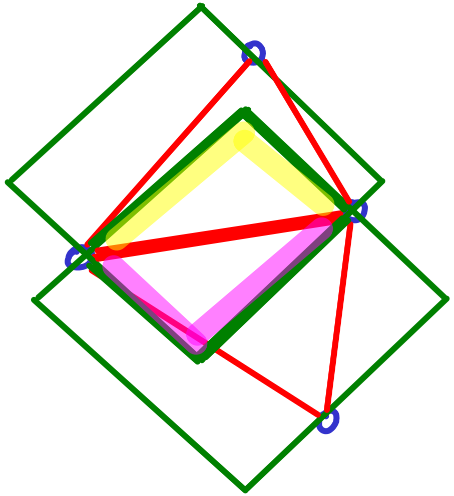


Circle graph

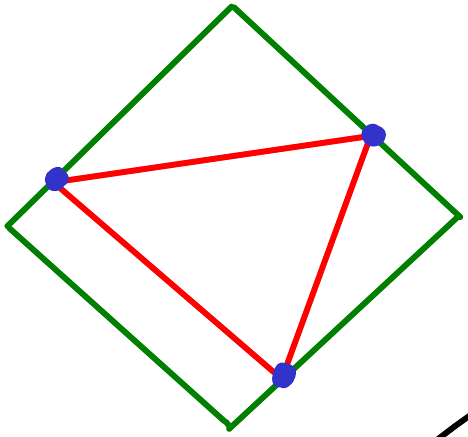




Each edge in T_{L_1}
contributes
2 circle graph
edges

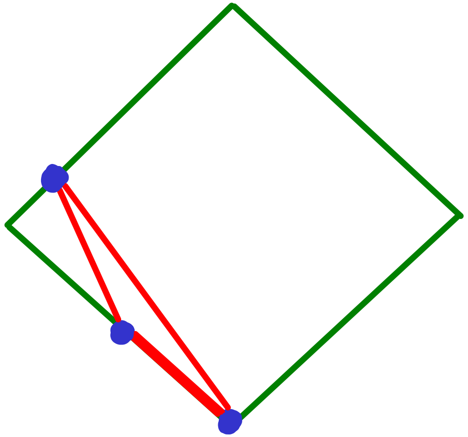


...except for exterior face edges



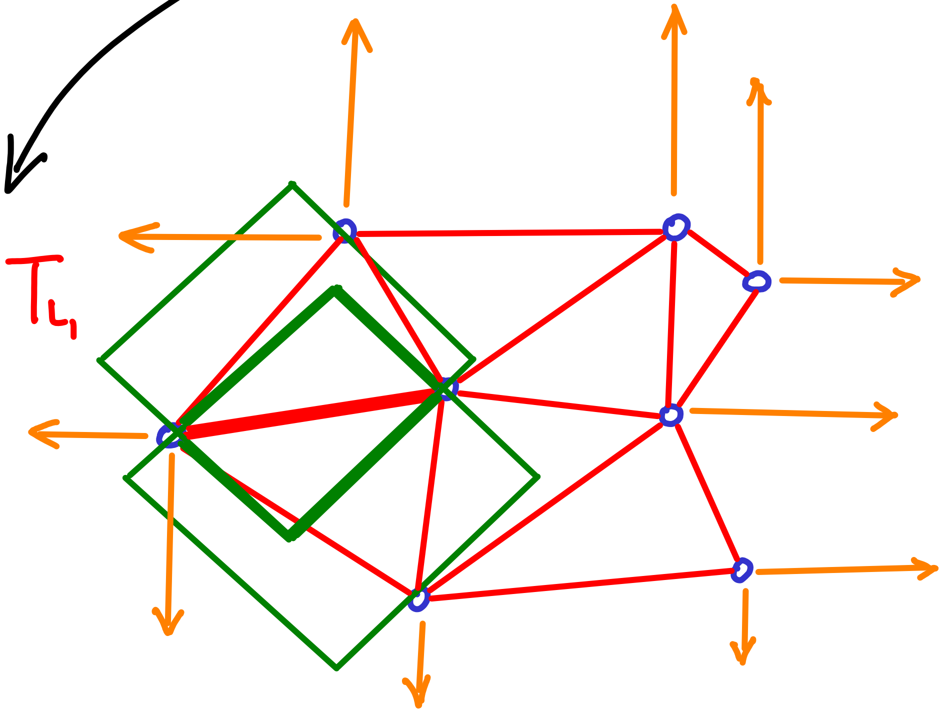
we have augmented the graph

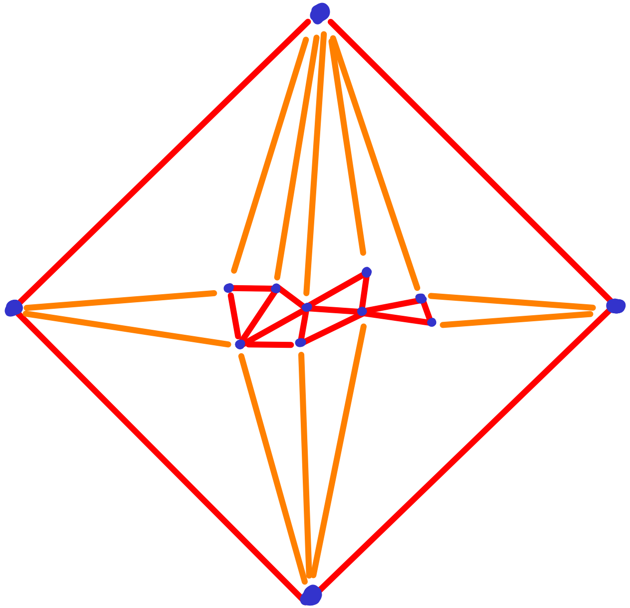
so every original edge has this property

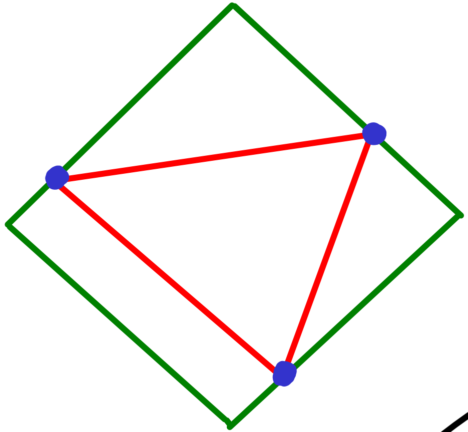


Each edge in T_{L_1} contributes

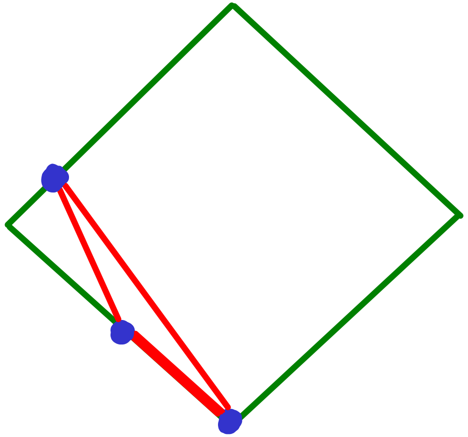
2 circle graph edges

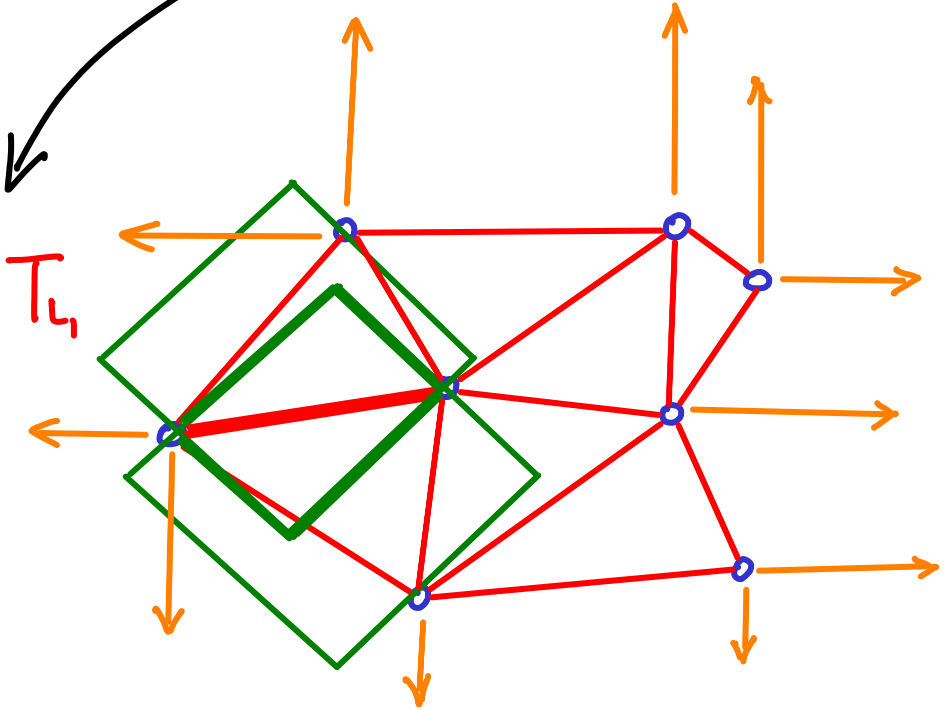






we have augmented the graph
so every original edge has this property



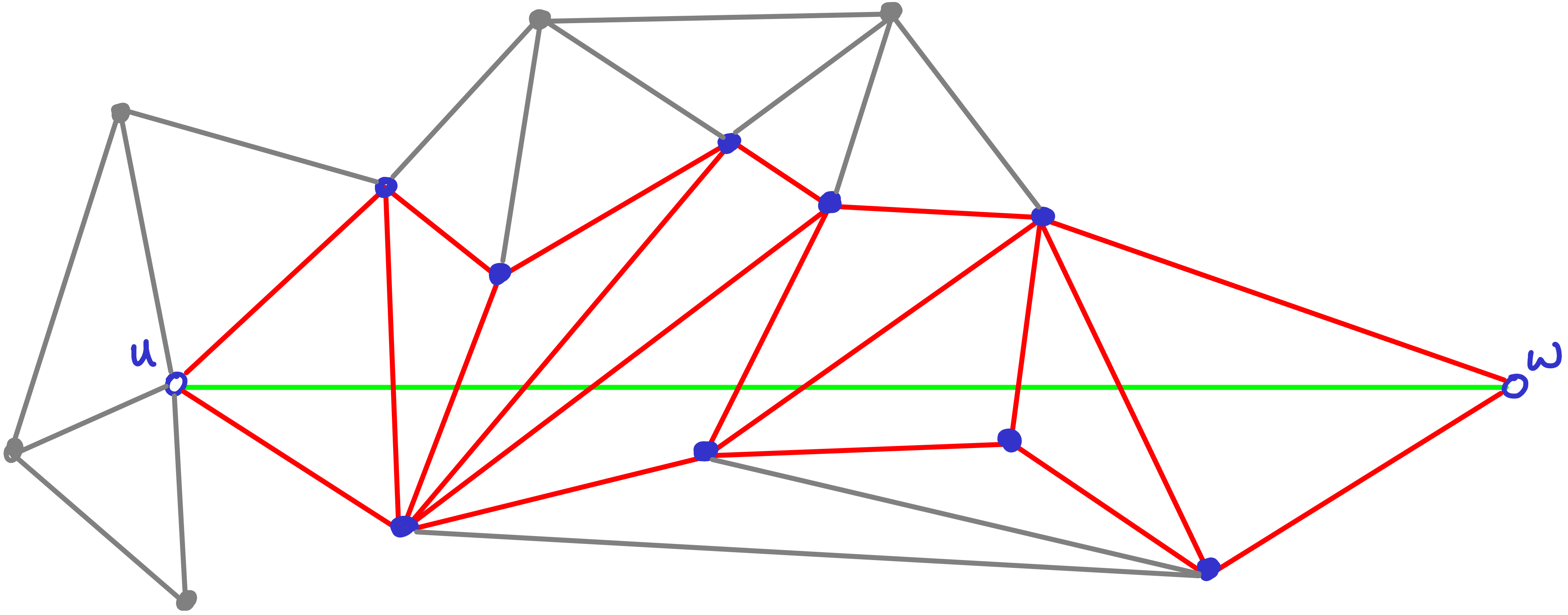


Each edge in T_{L_1} contributes 2 circle graph edges

we will find an $\sqrt{8}$-approximation path in the Circle graph

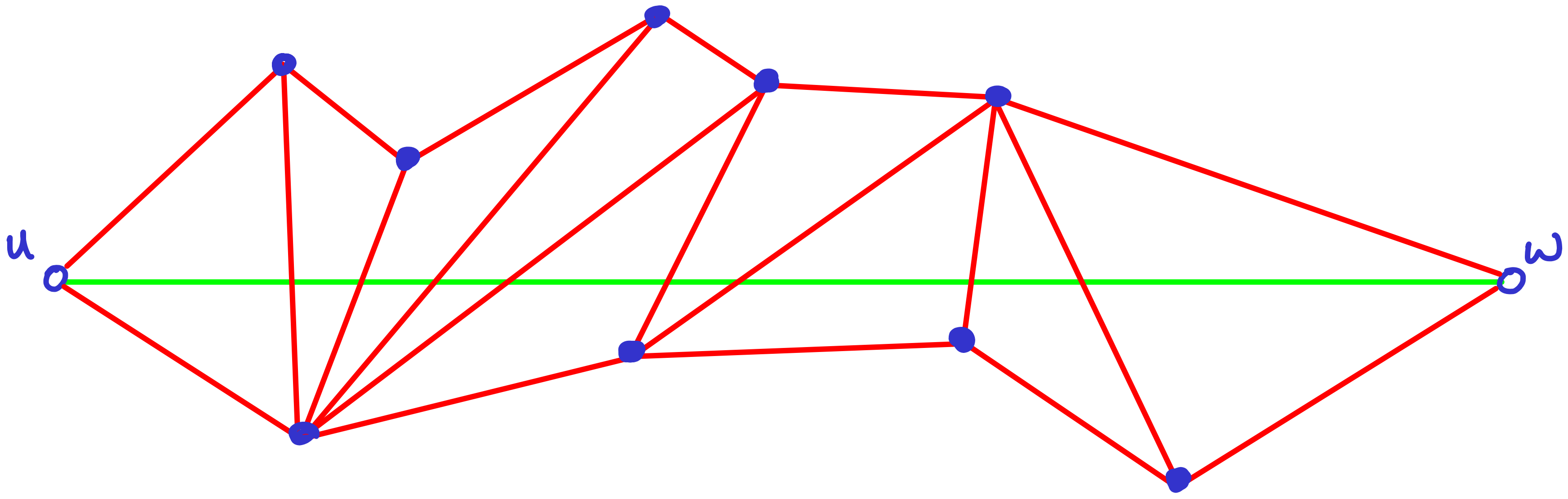
(using original edges clearly gives shortcuts)

We only need the subgraph of triangles that contain part of $\overline{uw}$



We only need the subgraph of triangles that contain part of $\overline{uw}$

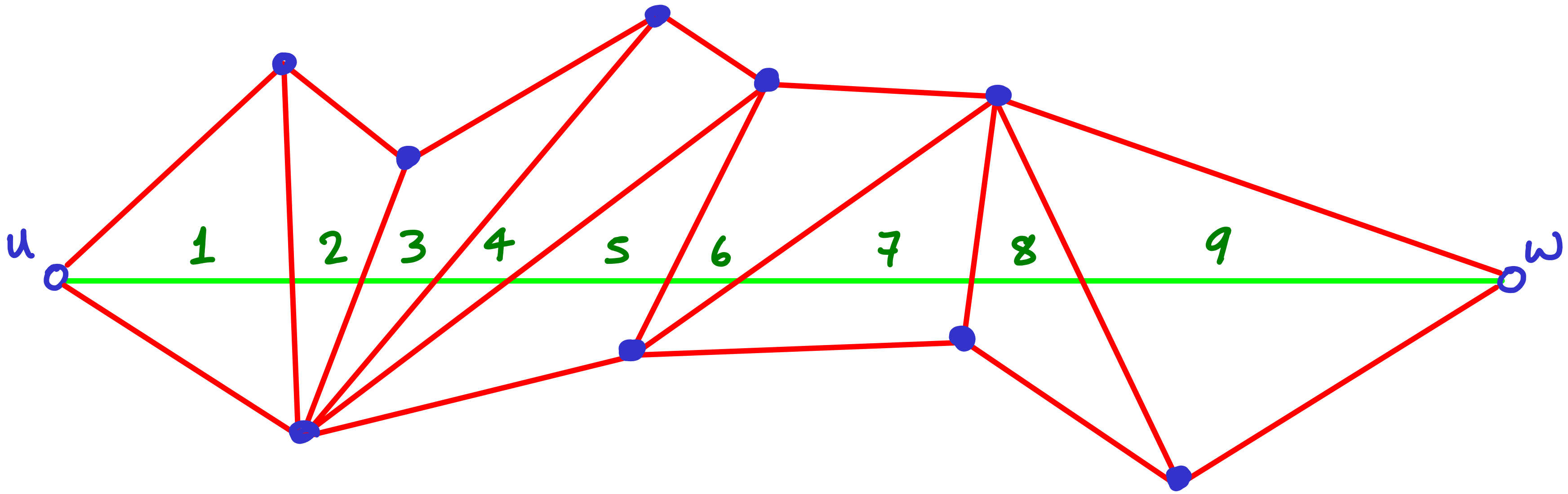
The path that approximates $\overline{uw}$ will only use edges of this subgraph.
(circle graph)



We only need the subgraph of triangles that contain part of $\overline{uw}$

The path that approximates $\overline{uw}$ will only use edges of this subgraph.

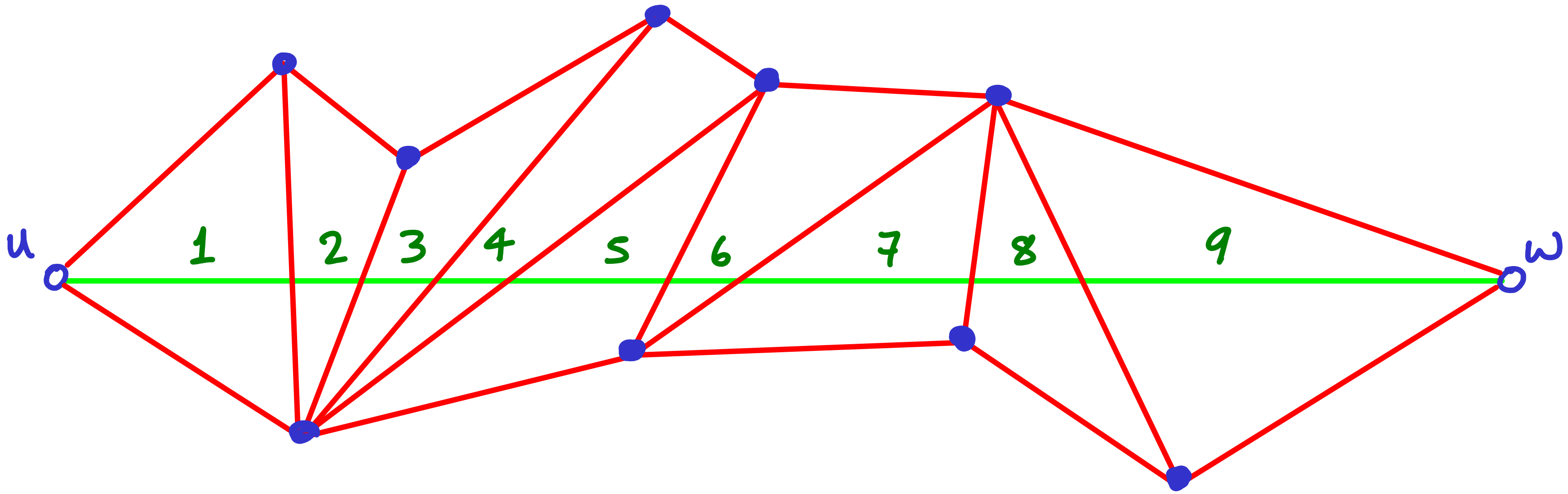
↳ in fact it will make steady progress, visiting triangles in order.



We only need the subgraph of triangles that contain part of $\overline{uw}$

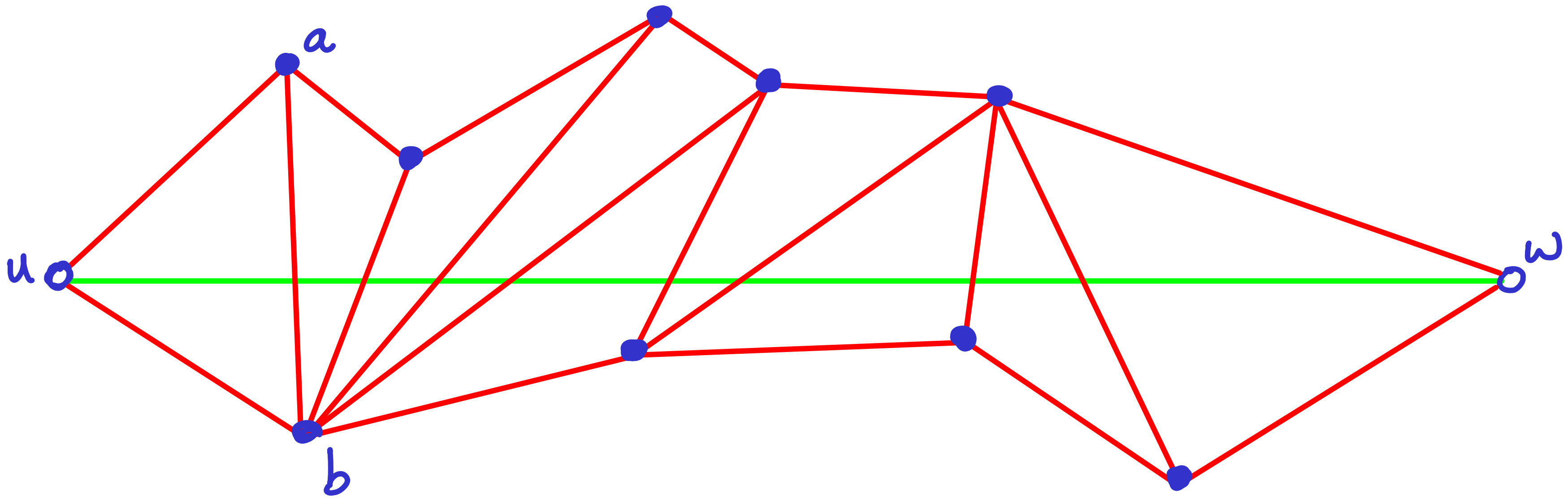
The path that approximates $\overline{uw}$ will only use edges of this subgraph.

↳ in fact it will make steady progress, visiting triangles in order.

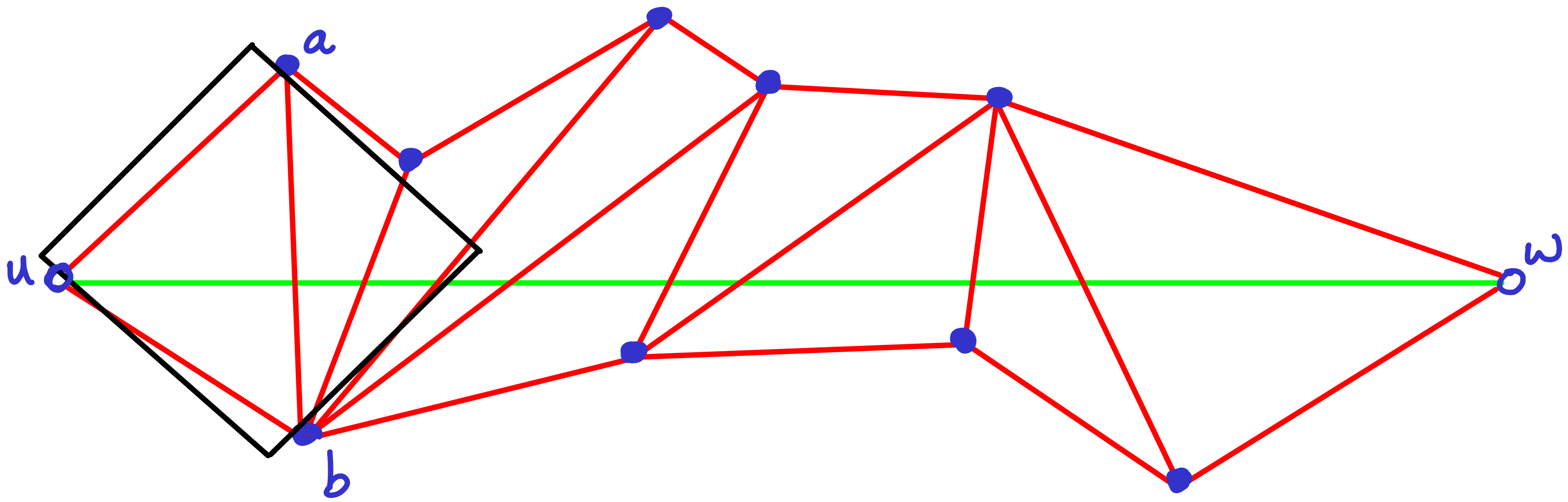


Assume no vertex on $\overline{uw}$ otherwise recurse

u belongs only to 1 triangle uab, with a above & b below $\overline{uw}$

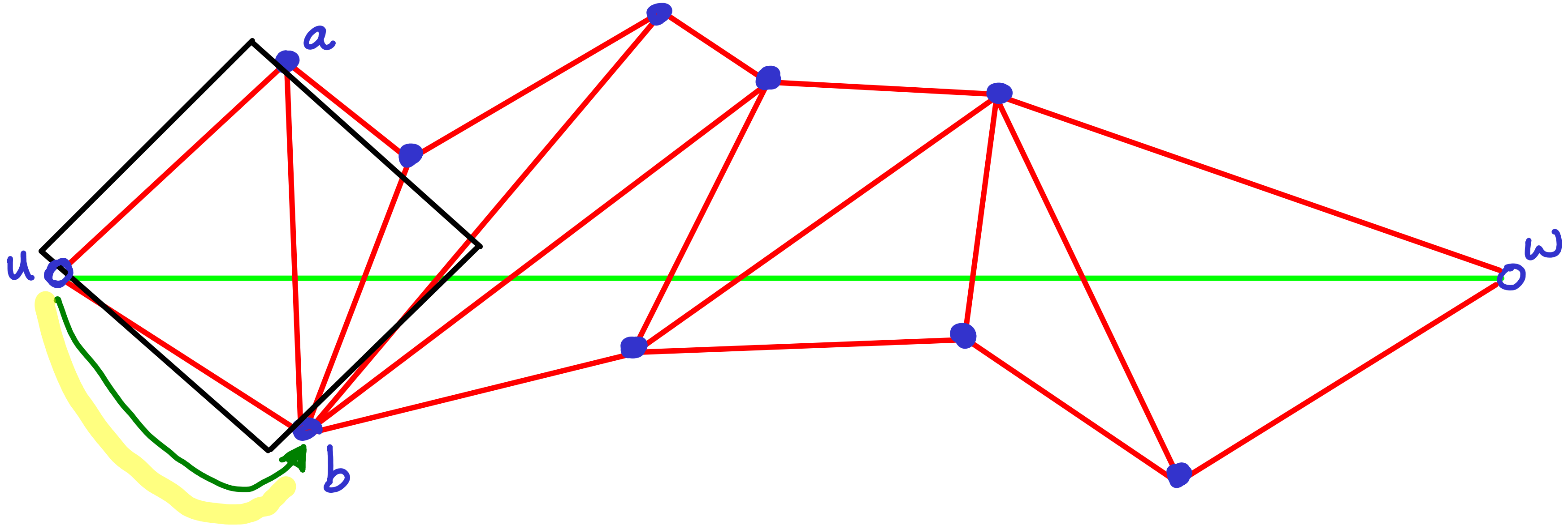


u belongs only to 1 triangle uab, with a above & b below $\overline{uw}$

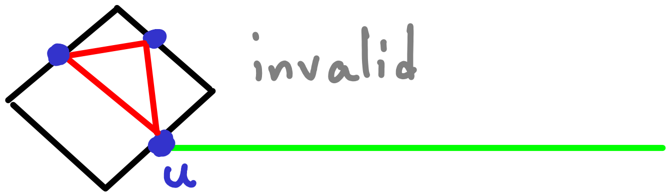


In fact u is on the or side of the empty diamond on uab.

u belongs only to 1 triangle uab, with a above & b below $\overline{uw}$

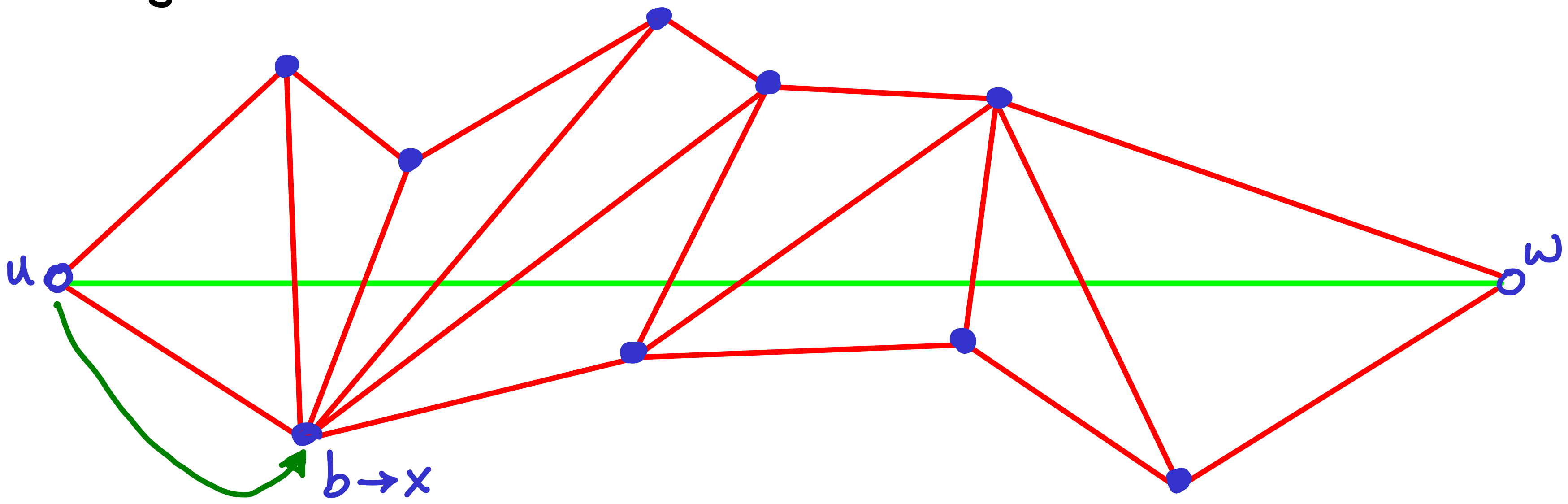


In fact u is on the ◇ or ◇ side of the empty diamond on uab.



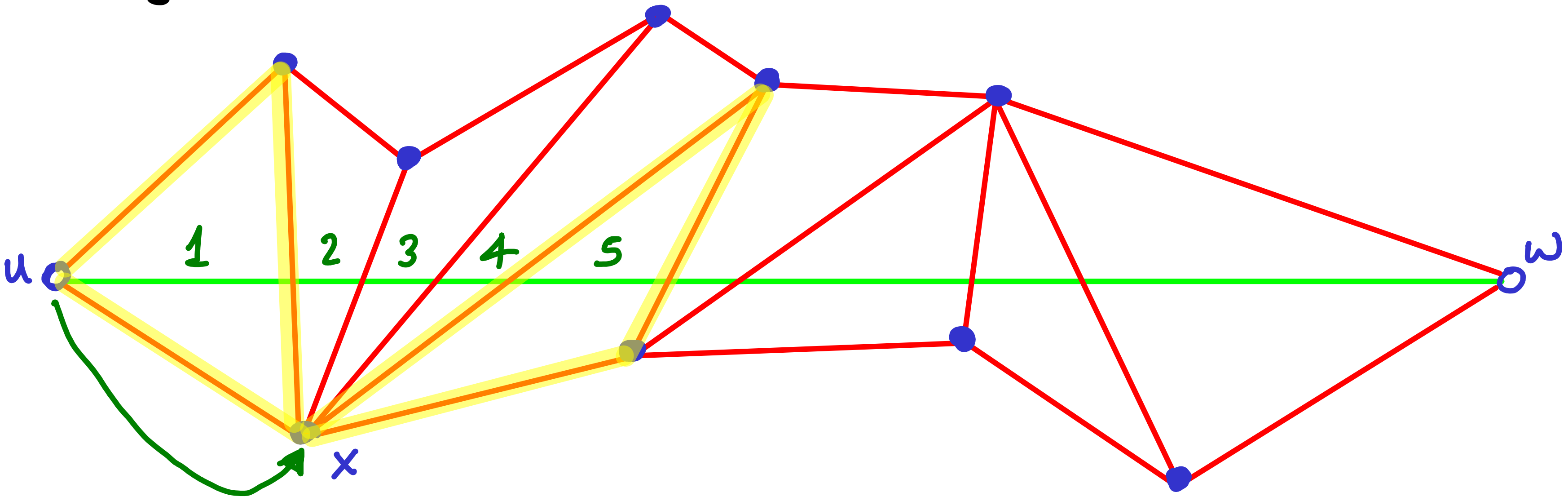
Start our path with u ↷ b [along ◇] because u is on ◇

$b \to x$ belongs to many triangles.



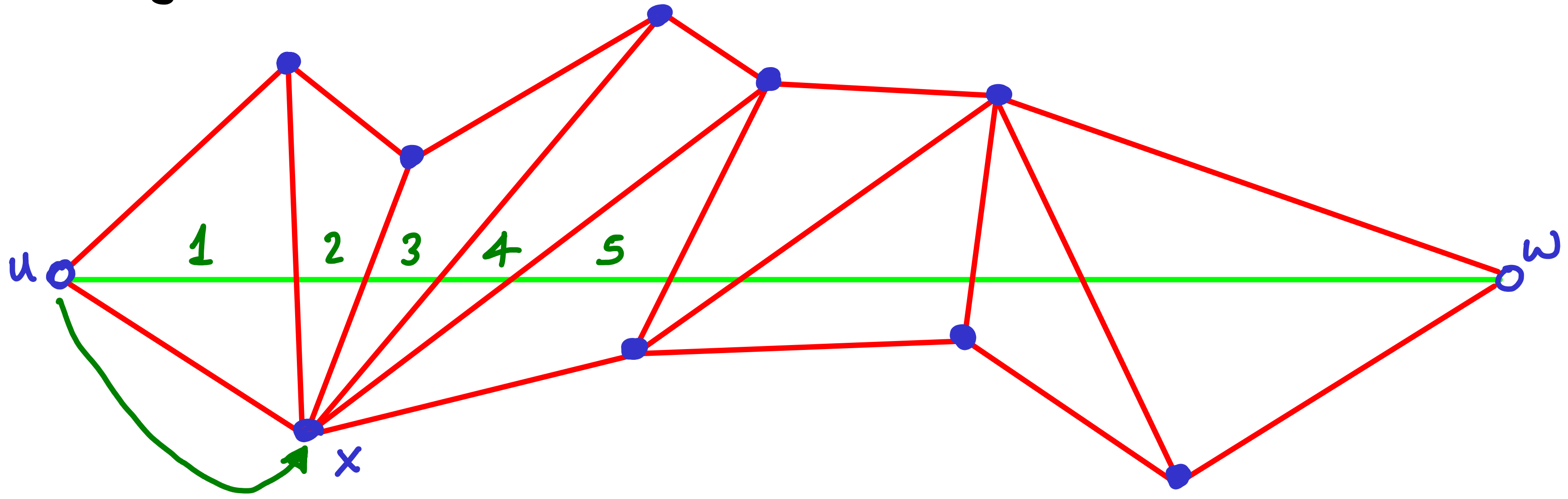
current vertex: x

x belongs to many triangles. We care about the rightmost one.

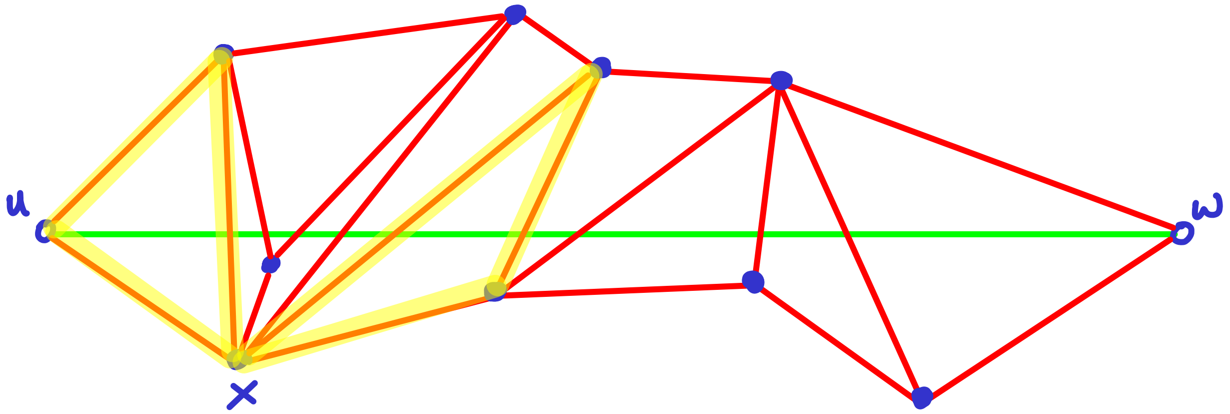


current vertex x

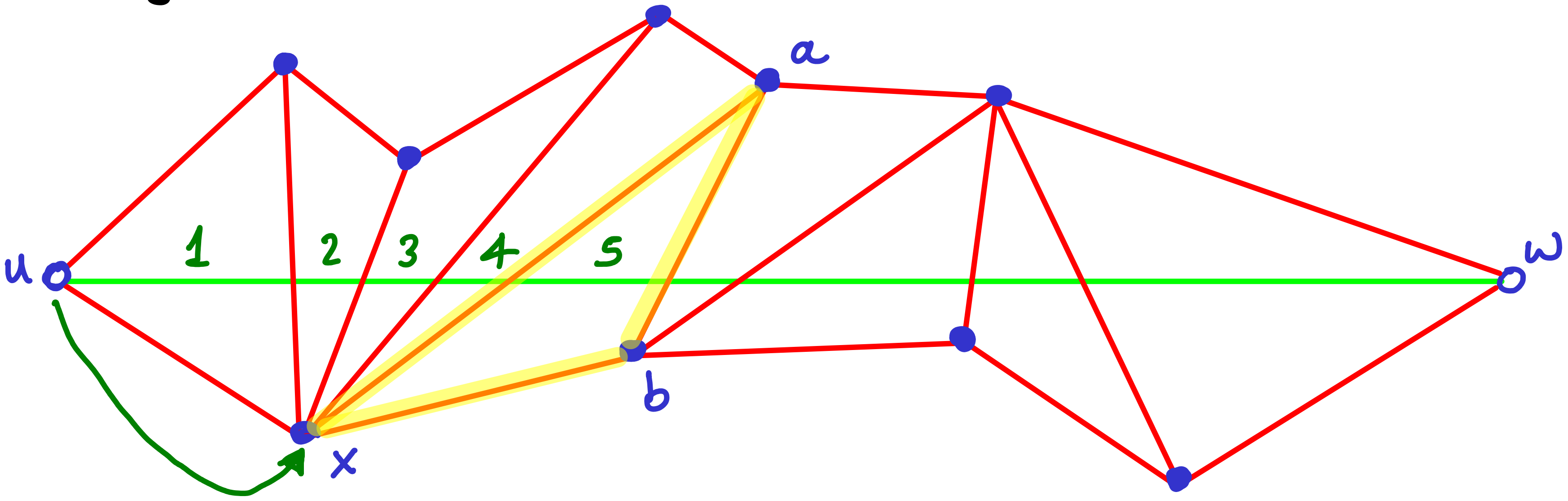
x belongs to many triangles. We care about the rightmost one.



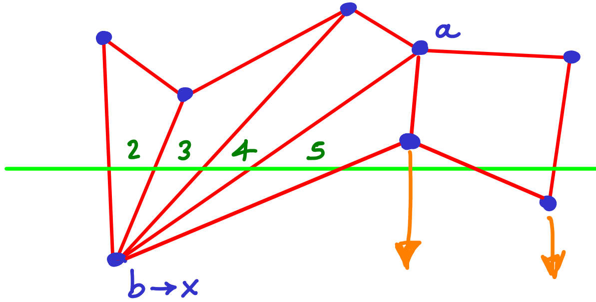
current vertex x



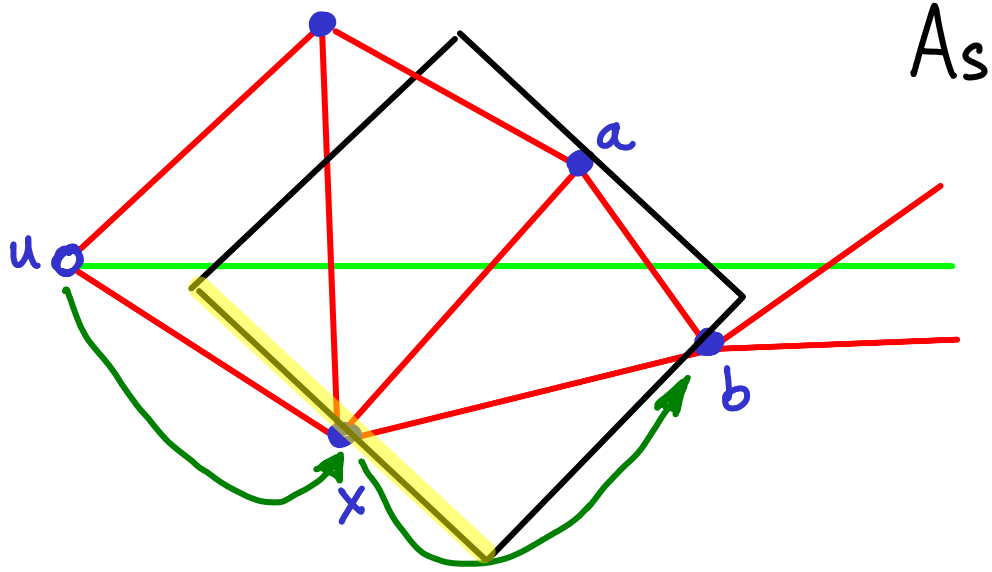
x belongs to many triangles. We care about the rightmost one.

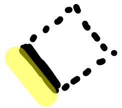
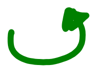


Again we have the property that the current vertex x on our path is in a triangle xab, w/ a above & b below uw



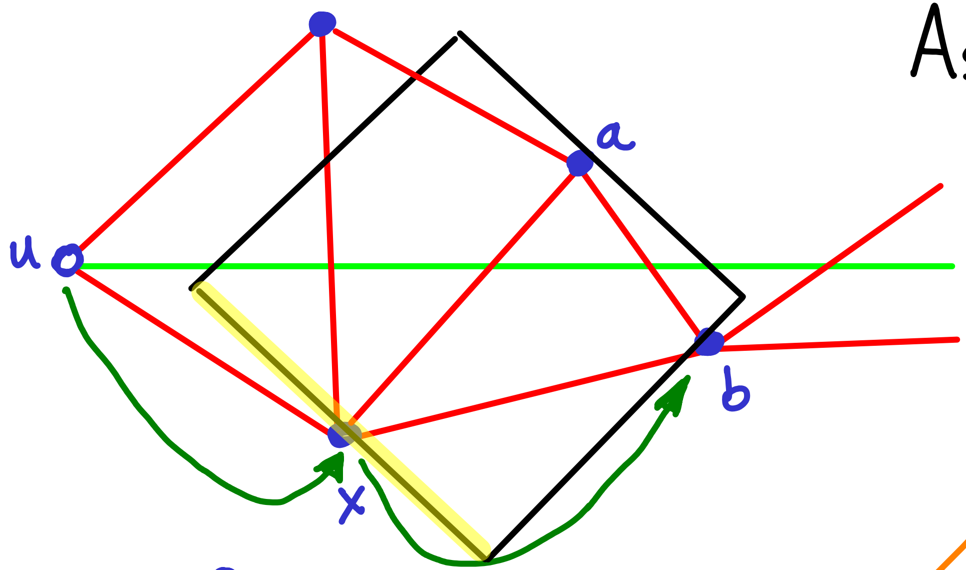
fake vertices help

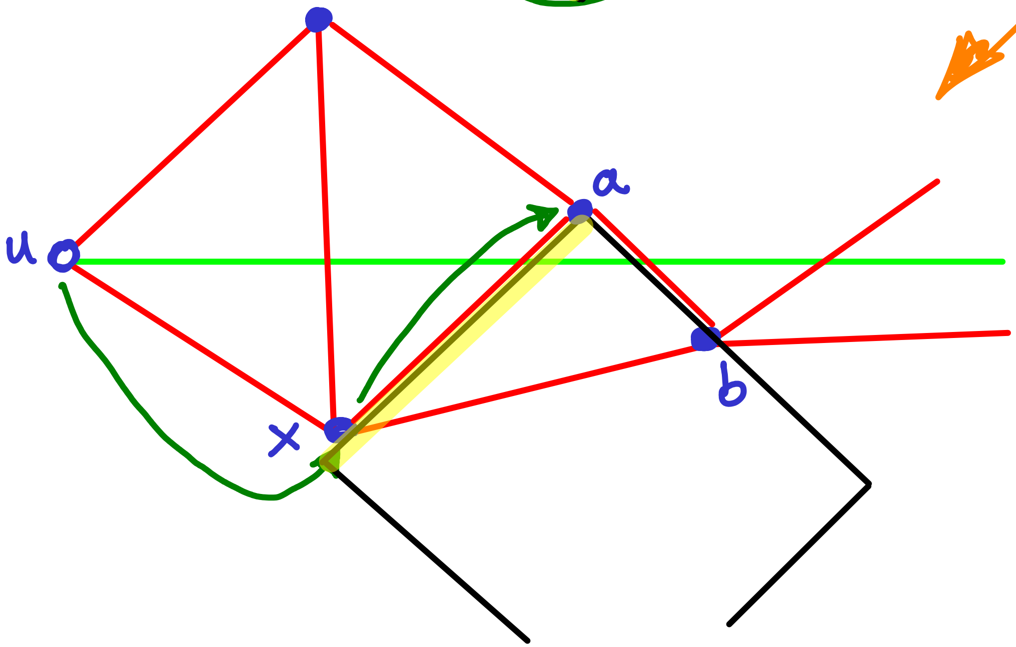


As before, if x is on  move 

As before, if x is on move

else if x is on move





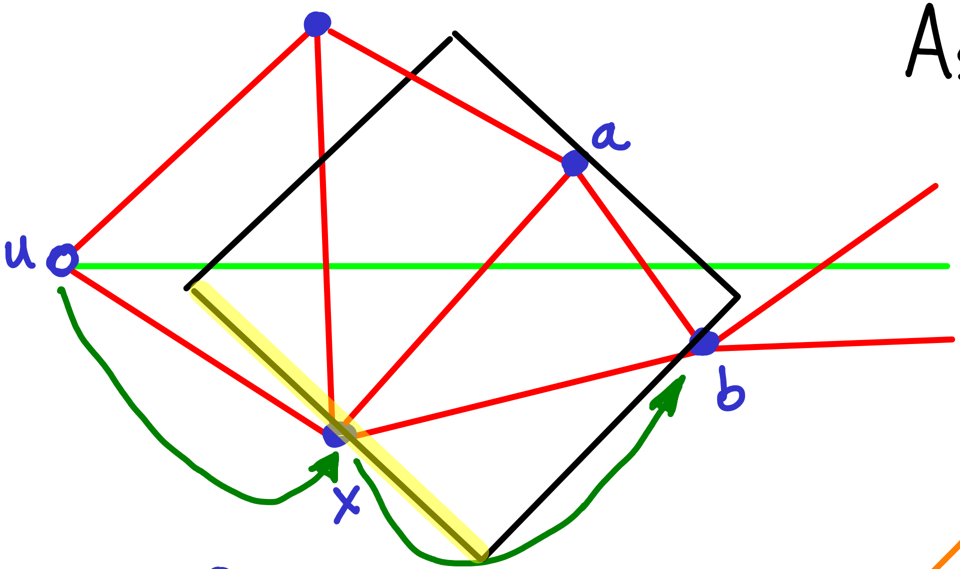
As before, if x is on move ↺

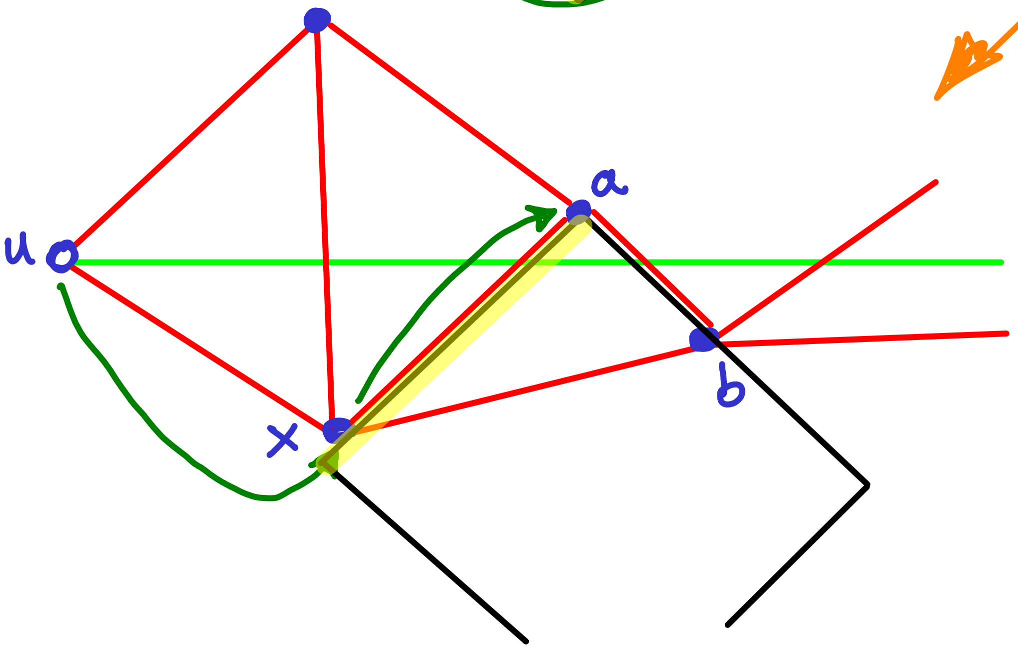
else if x is on move ↻

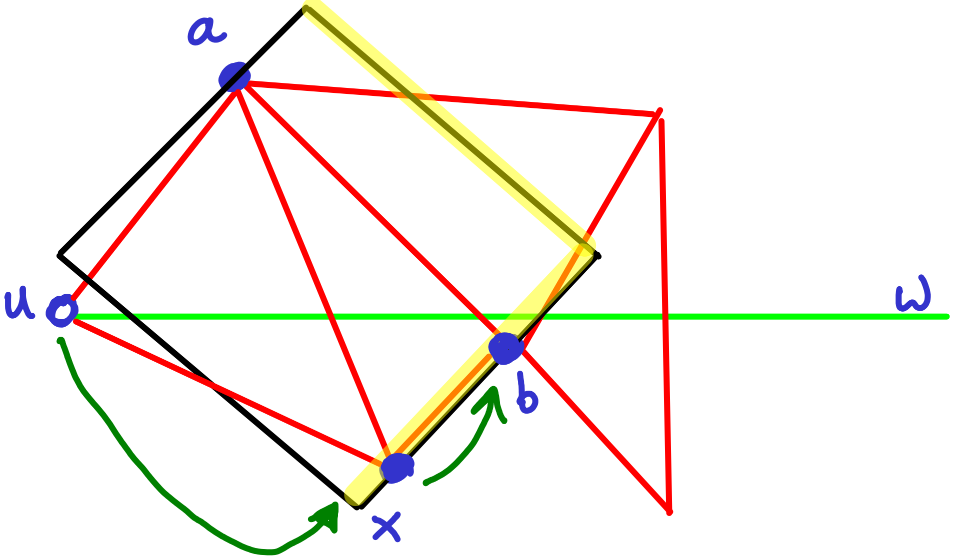
<u>else</u> // x is on

if x is above $\overline{uw}$ move ↻

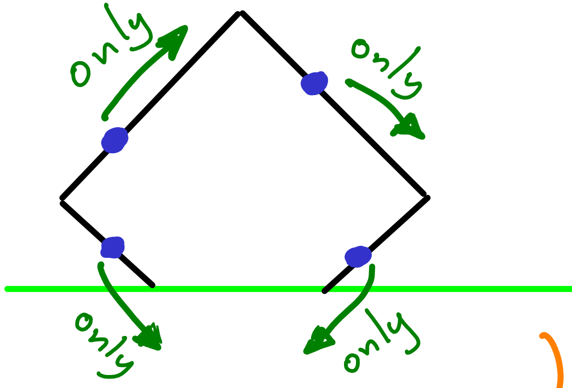
if x is below $\overline{uw}$ move ↺



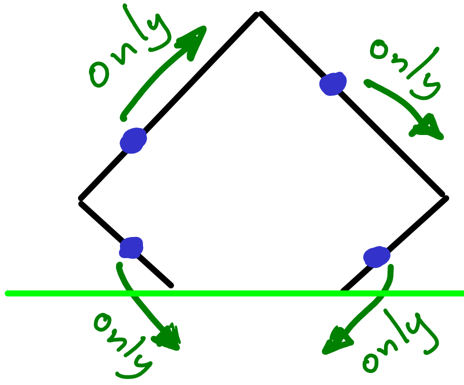


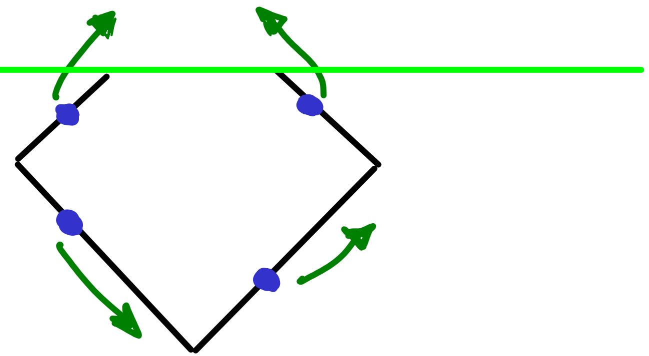


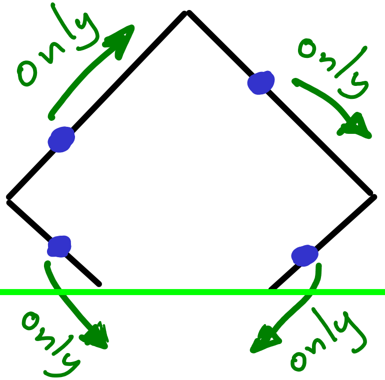
when moving above $\overline{uw}$



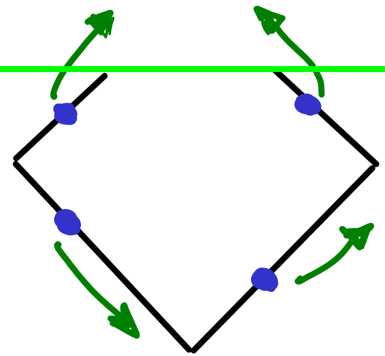
wherever you start, direction is defined when moving above $\overline{uw}$





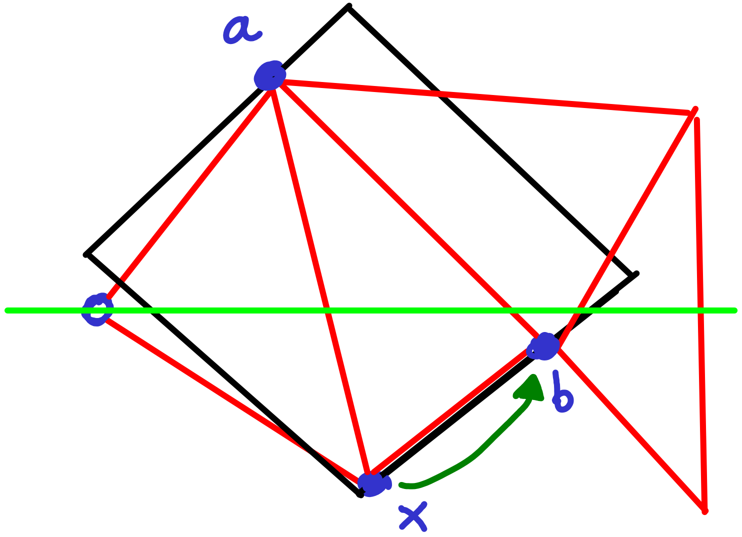


when moving above $\overline{uw}$



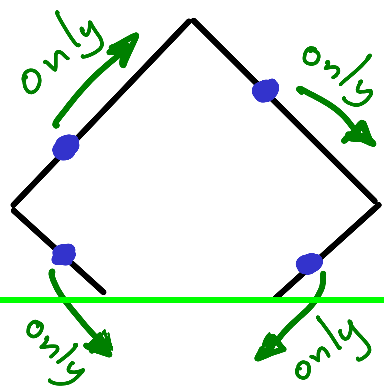
When can we switch sides?

e.g below → above



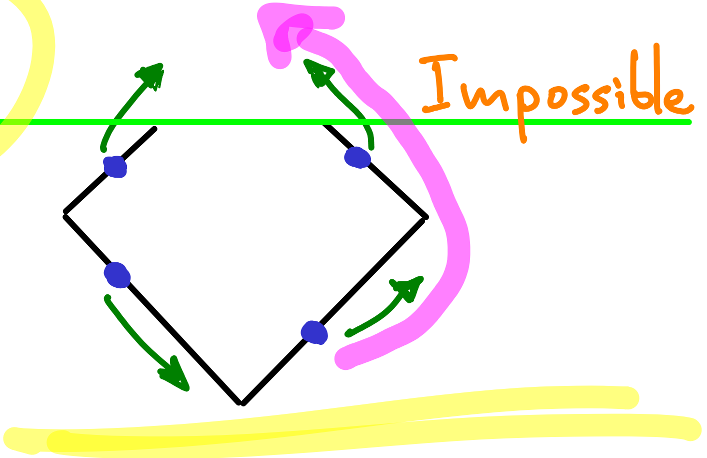
Notice that b must be below $\overline{uw}$ if $\times$ is on

otherwise we are not looking at the rightmost triangle of $\times$



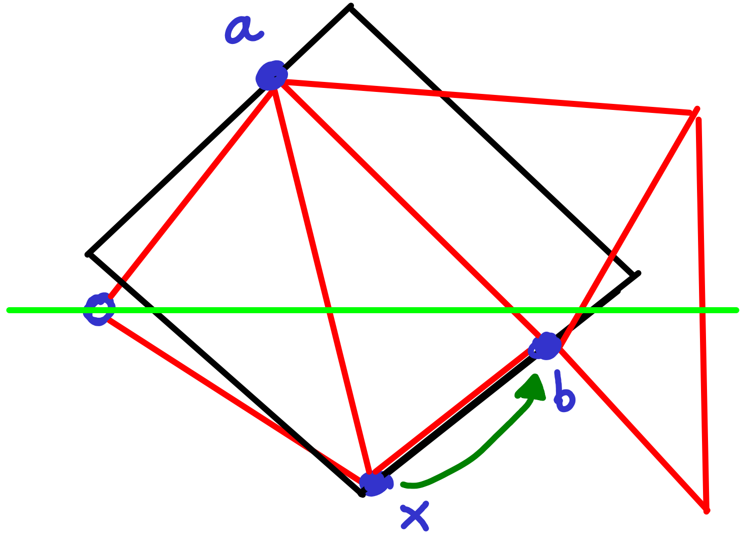
when moving above $\overline{uw}$

can't cross over like this



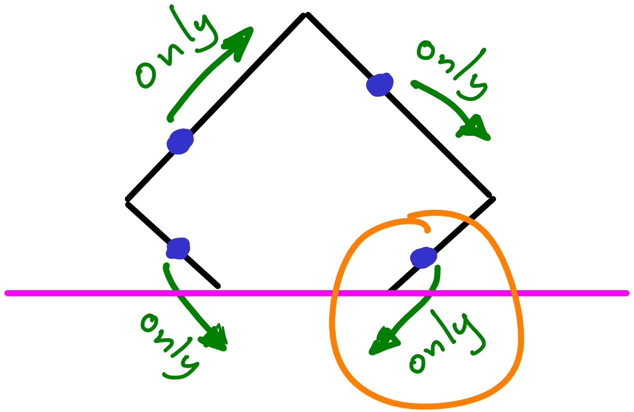
When can we switch sides?

e.g below → above



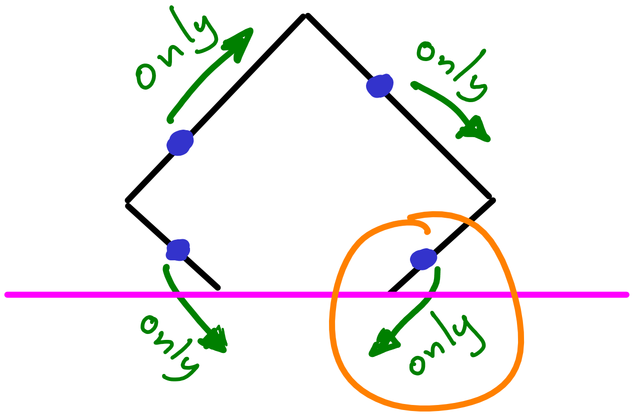
Notice that b must be below $\overline{uw}$ if x is on 

otherwise we are not looking at the rightmost triangle of x

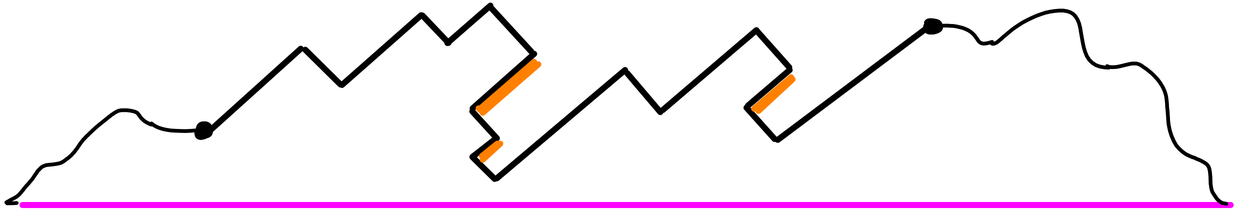


the only case where we move to the left

can't cross over, so we get to bound the backtracking

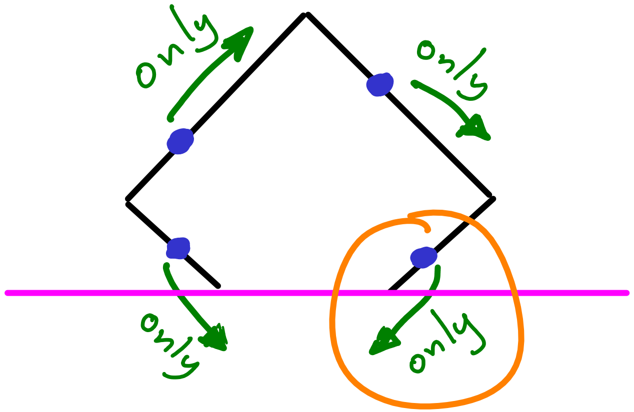


while above...

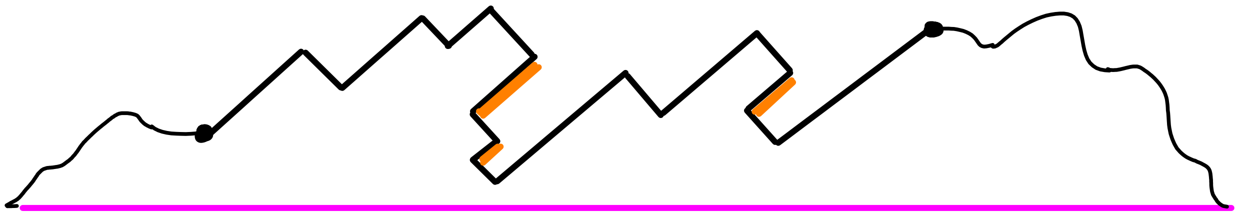


the only case where we move to the left

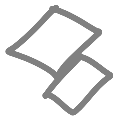
can't cross over, so we get to bound the backtracking



while above...

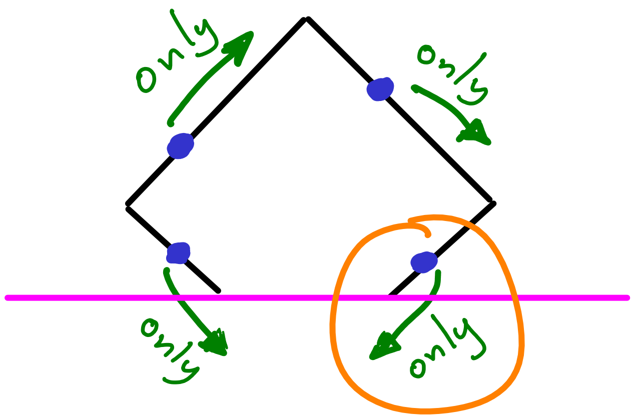


(why can't we switch from ↙ to ↗ ?)

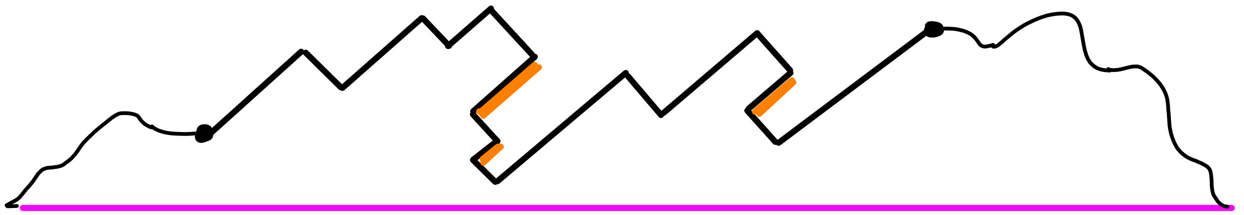
would involve placement of 

s.t. the corresponding triangles inside would not overlap — in proper order.

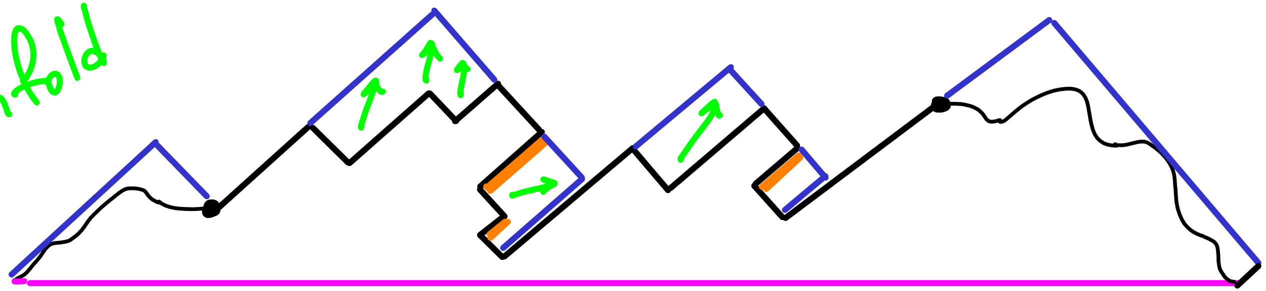
super-hand-wavy answer:

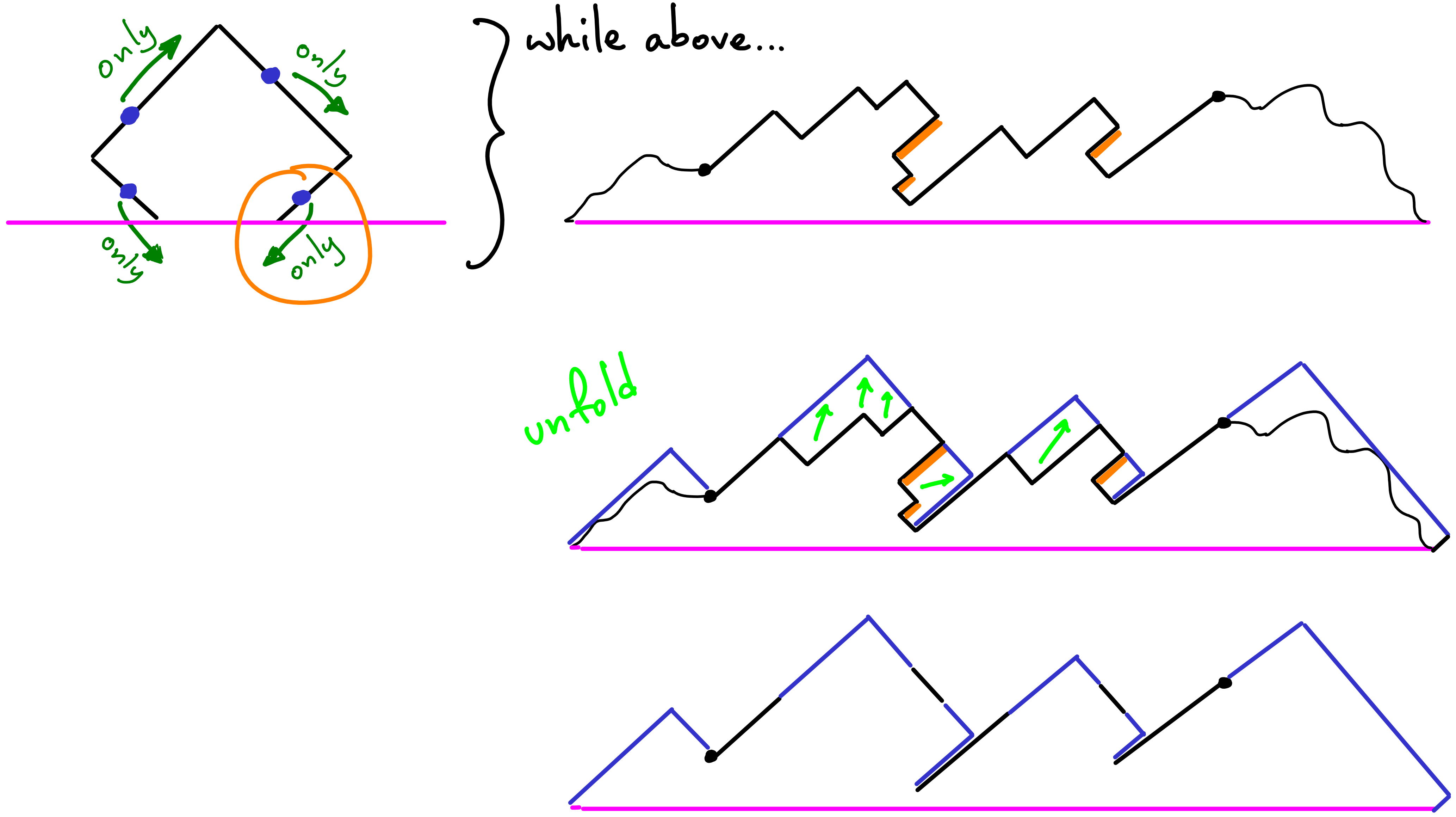


while above...

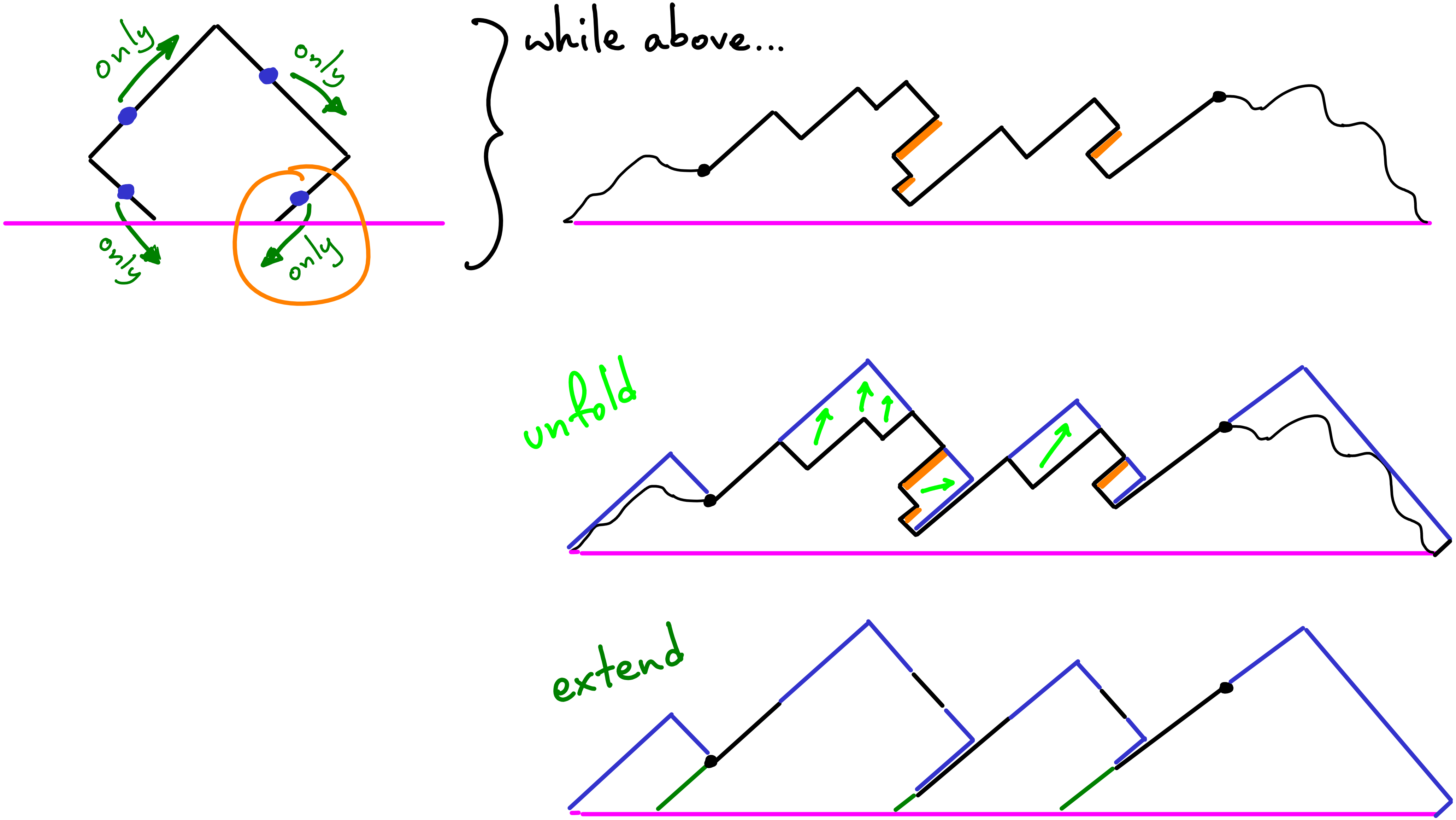


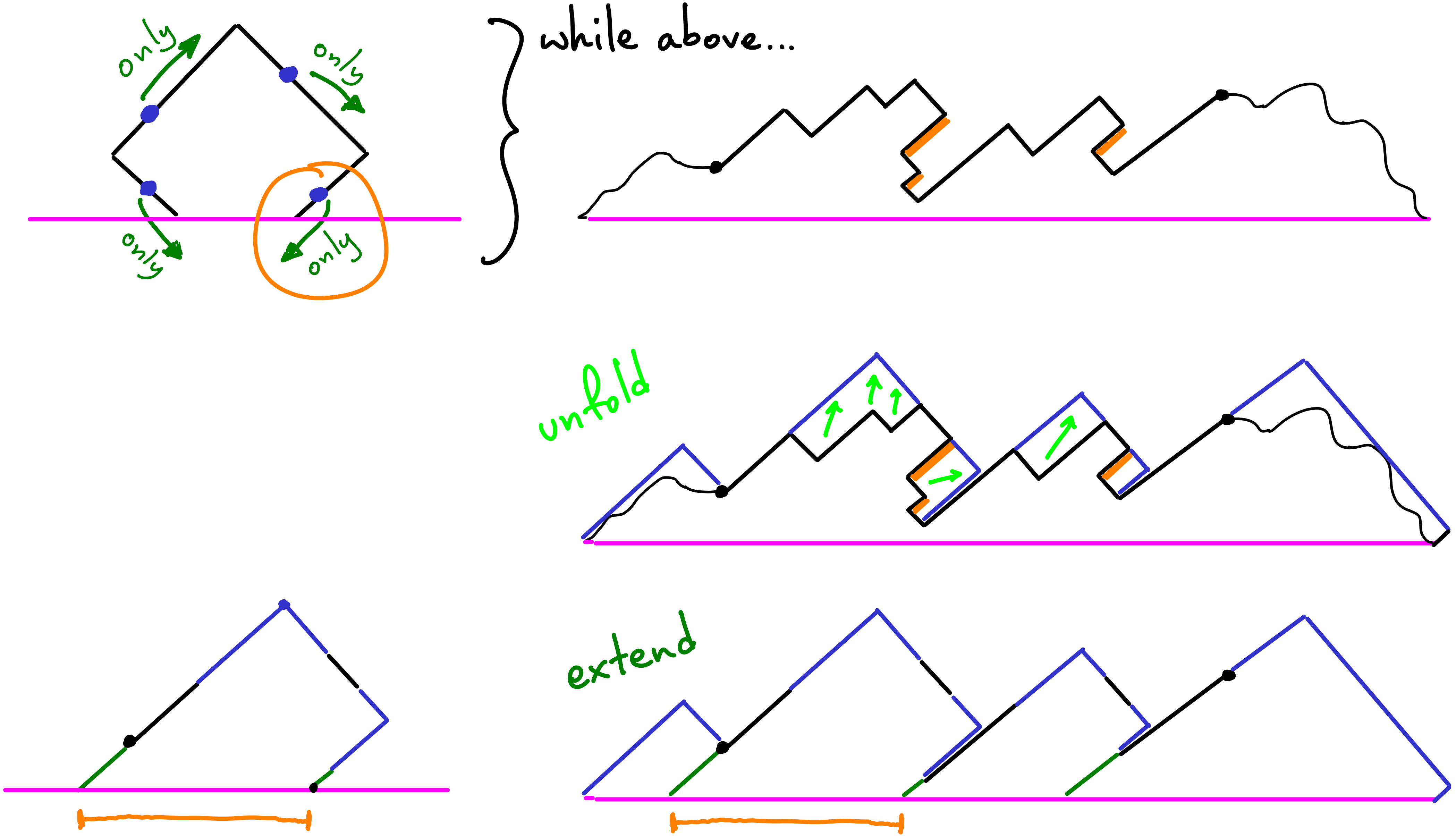
unfold

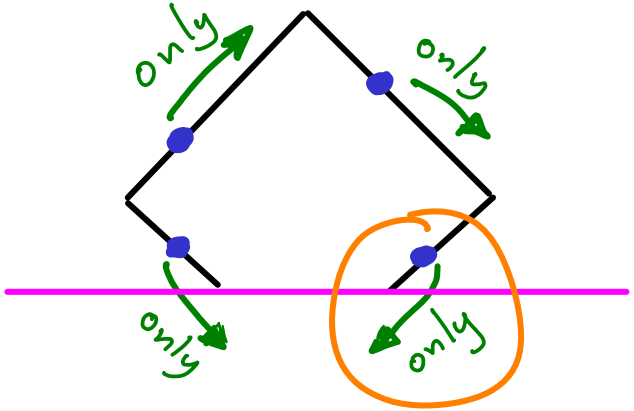




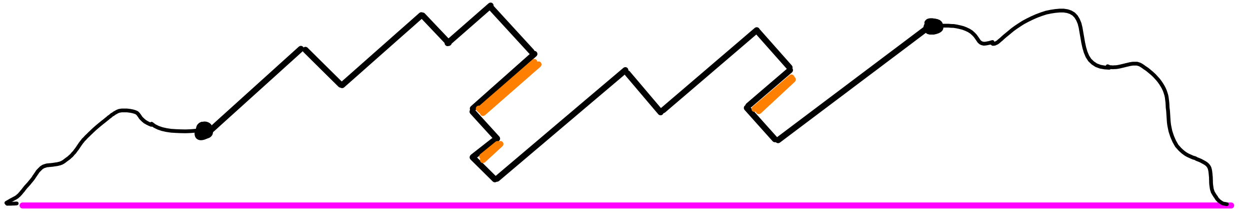
only
only
only
only
while above...
unfold


only
only
only
only
while above...
unfold
extend

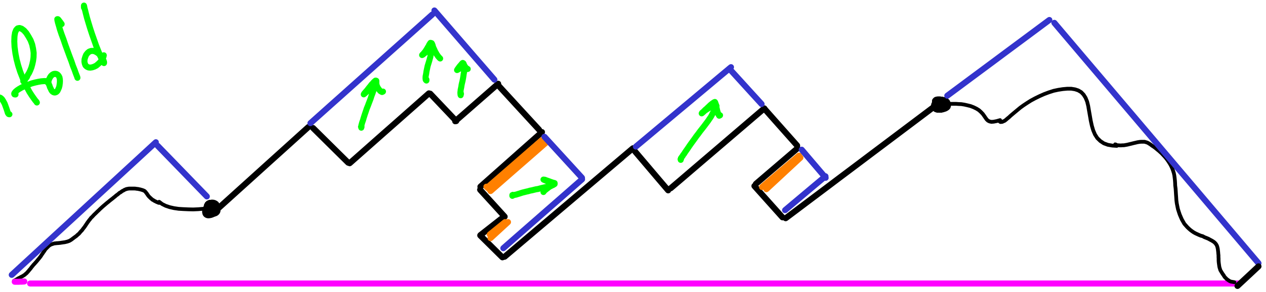

only
only
only
only
while above...
unfold
extend



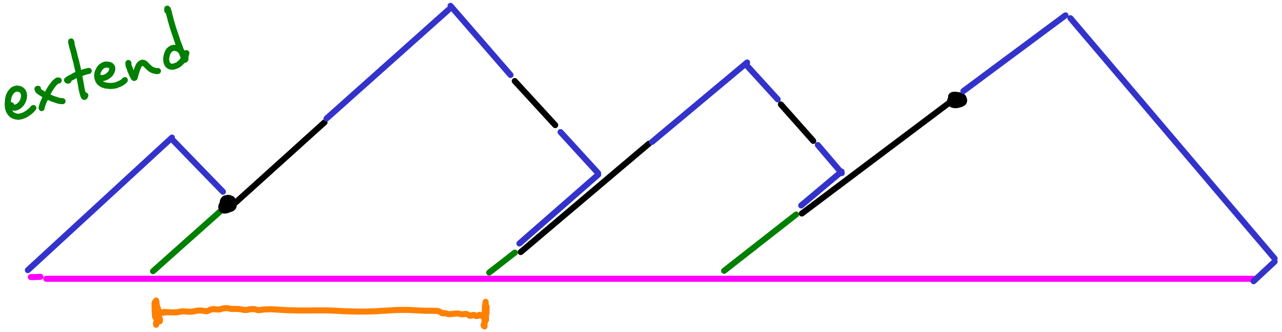
while above...



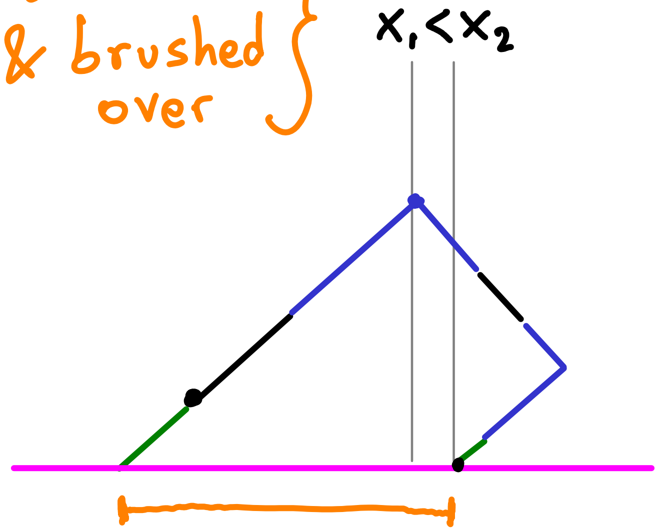
unfold

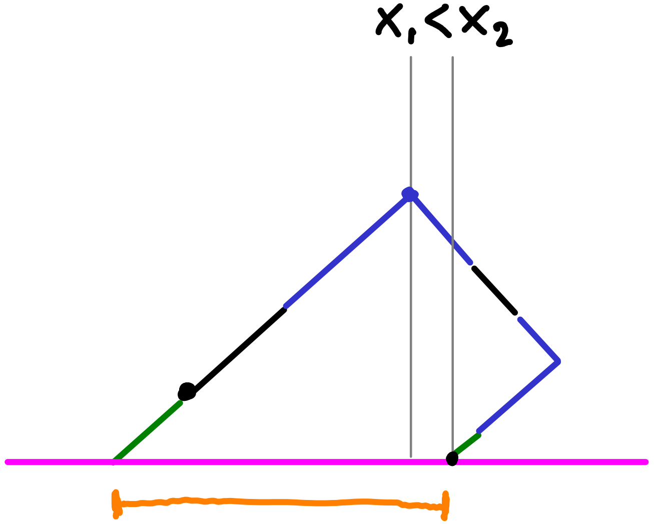


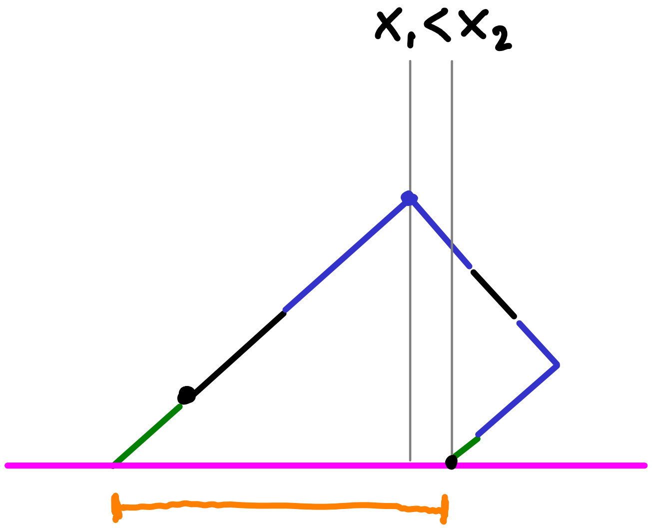
extend



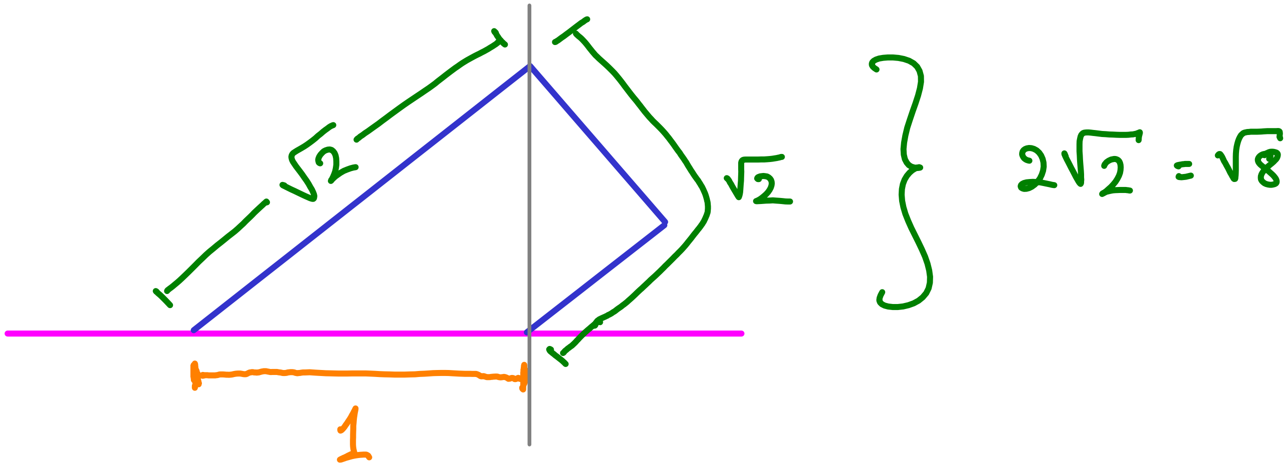
technical & brushed over

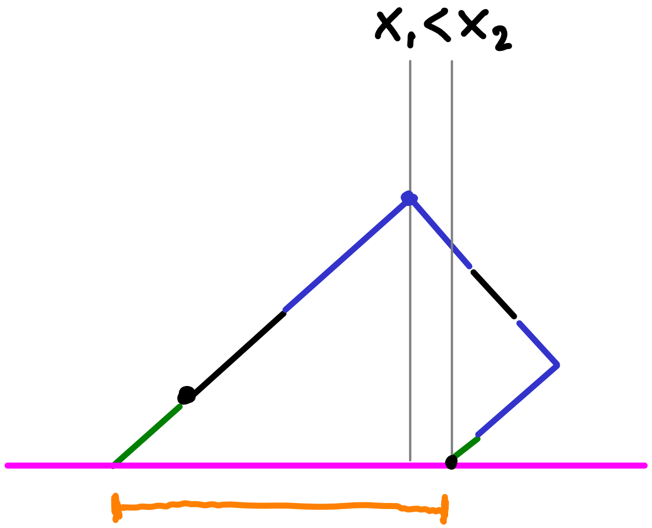



$x_1 < x_2$

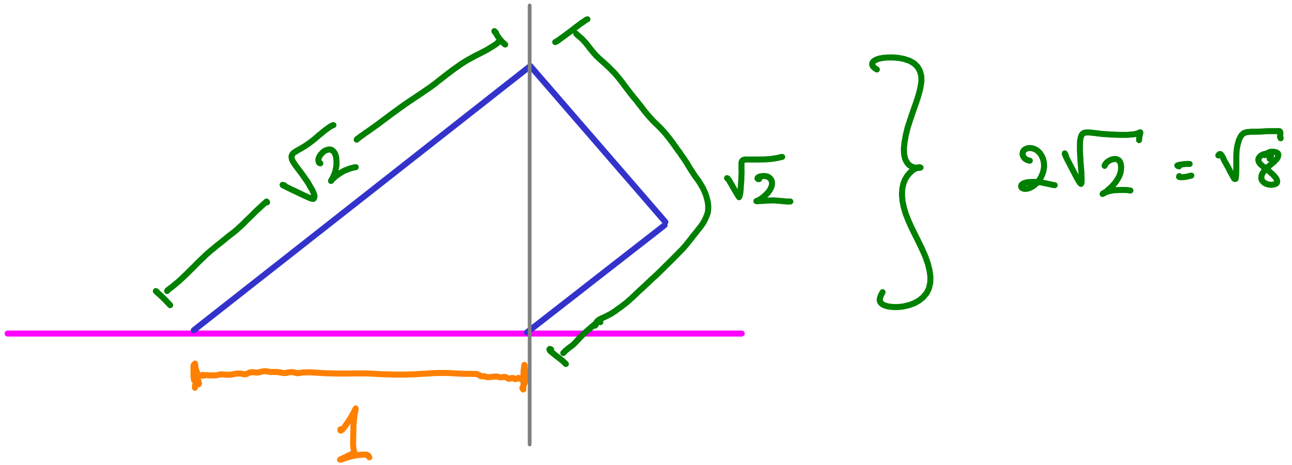


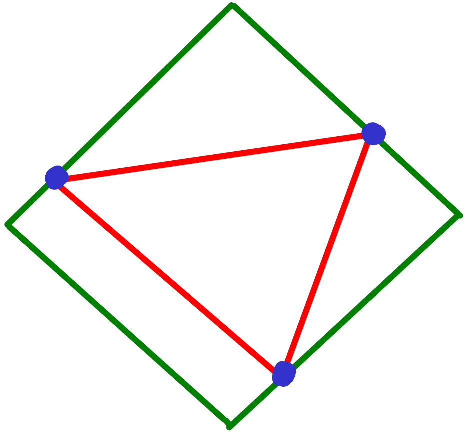
worst case

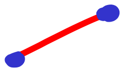


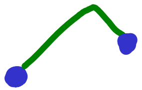


worst case

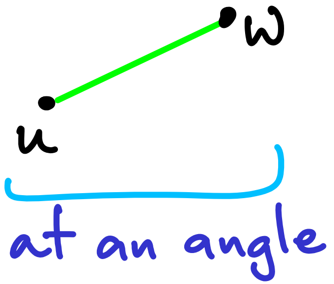




In fact we travel on 

not on 

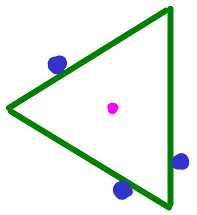
so the bound is better.

Dealing w/ 

is sort of skipped but claimed to give $\sqrt{10}$

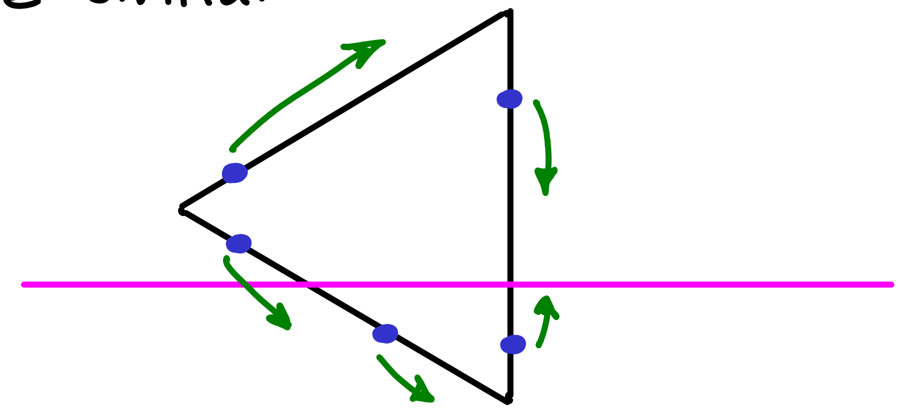
Computation : Delaunay triangulation (L_1 or L_2) : $\Theta(n \log n)$

Journal version contains improvement : 2-spanner (from $\sqrt{10}$)

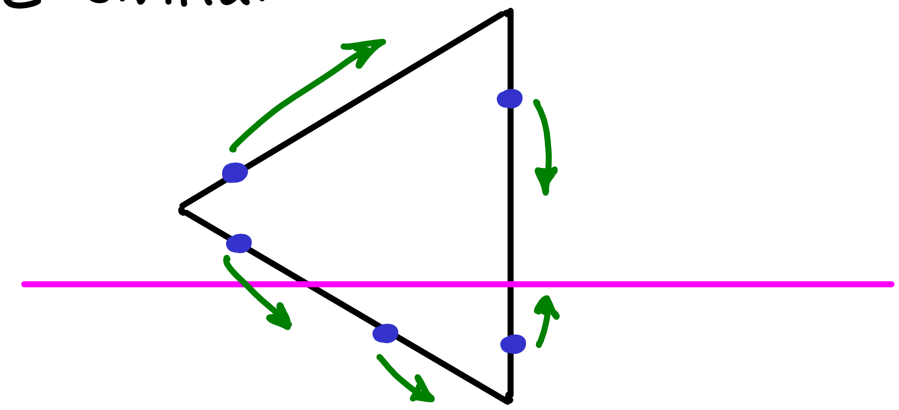


← empty circle

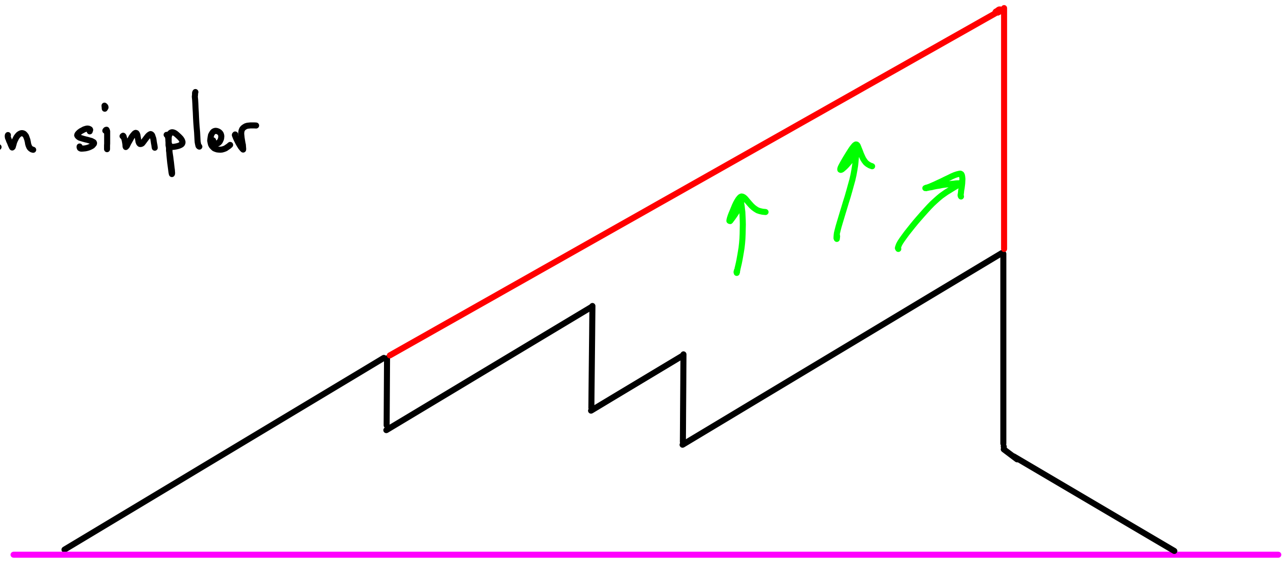
Rules are similar



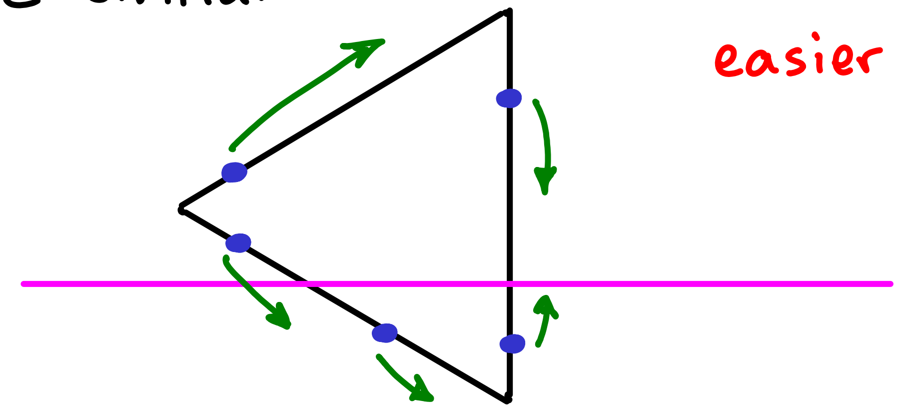
Rules are similar



Resulting shape is even simpler

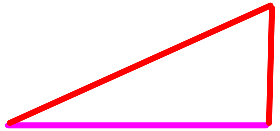


Rules are similar



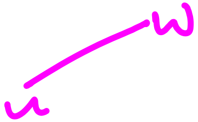
easier proofs ?

For horizontal uw

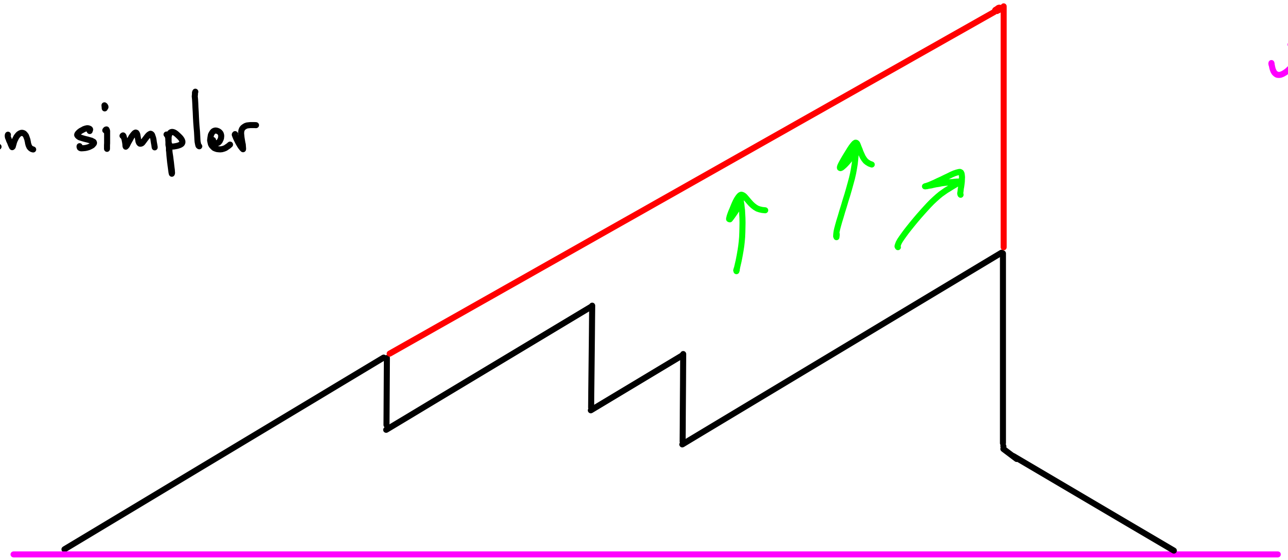


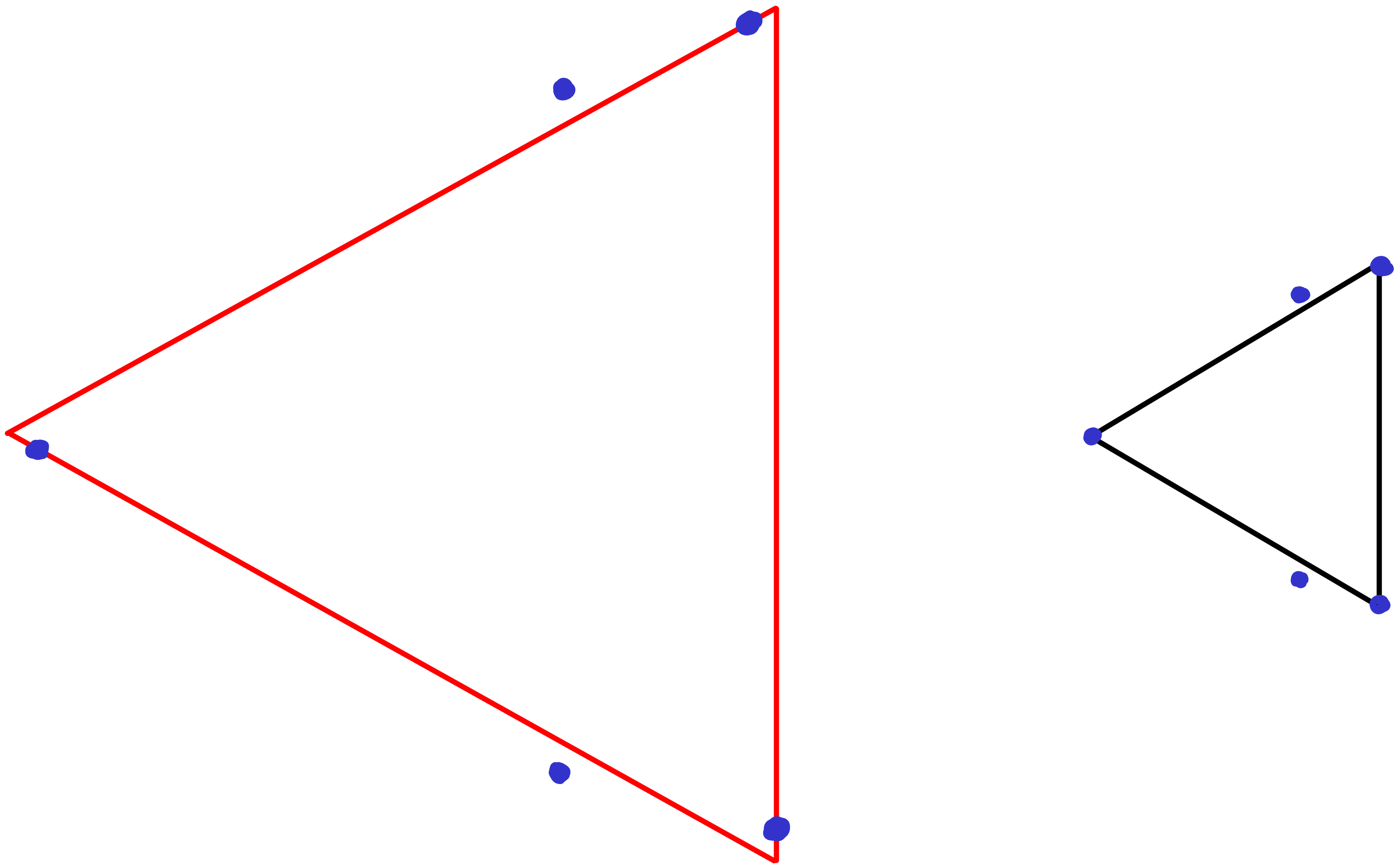
worst case ratio: $\sqrt{3}$

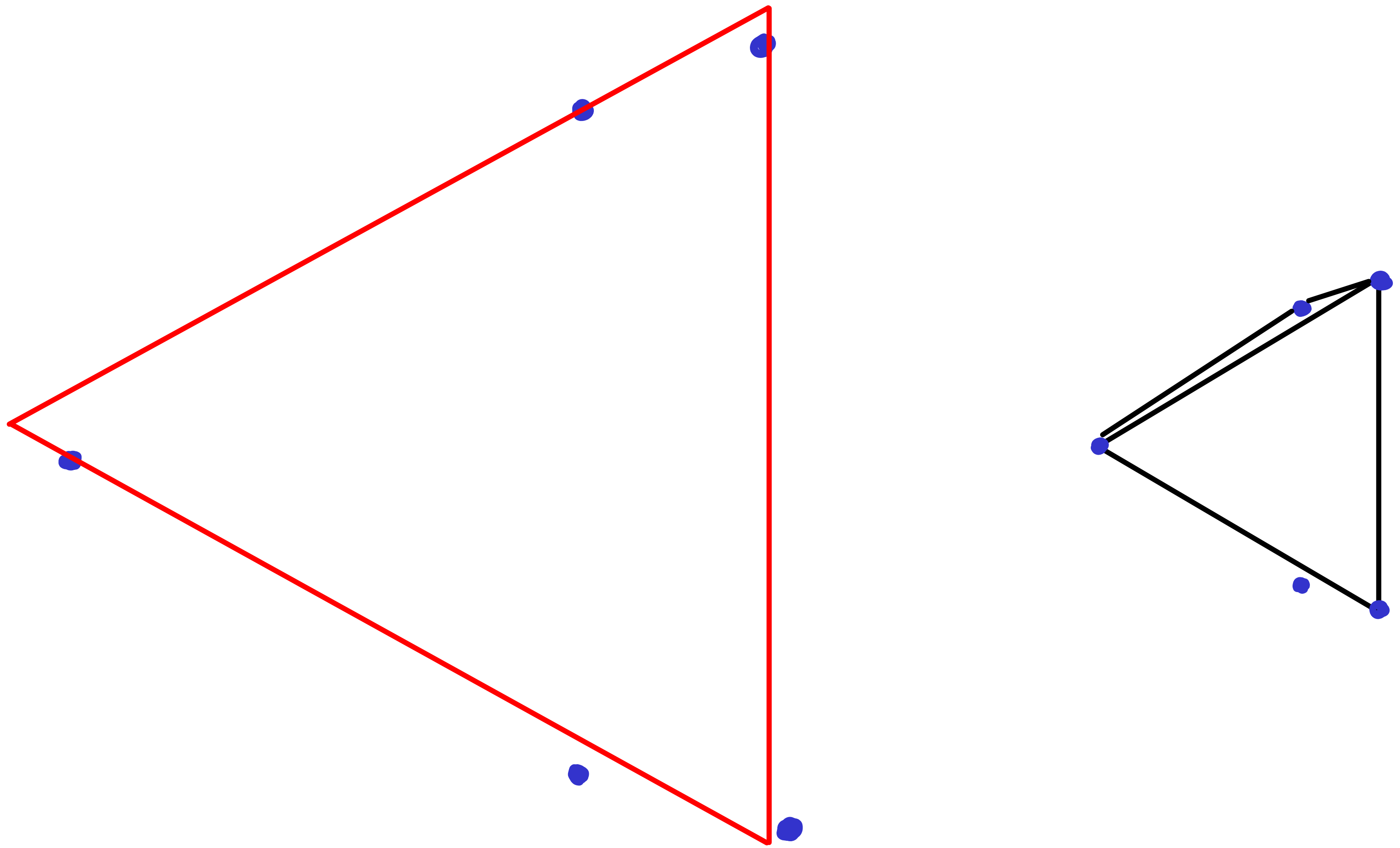
Becomes 2 for

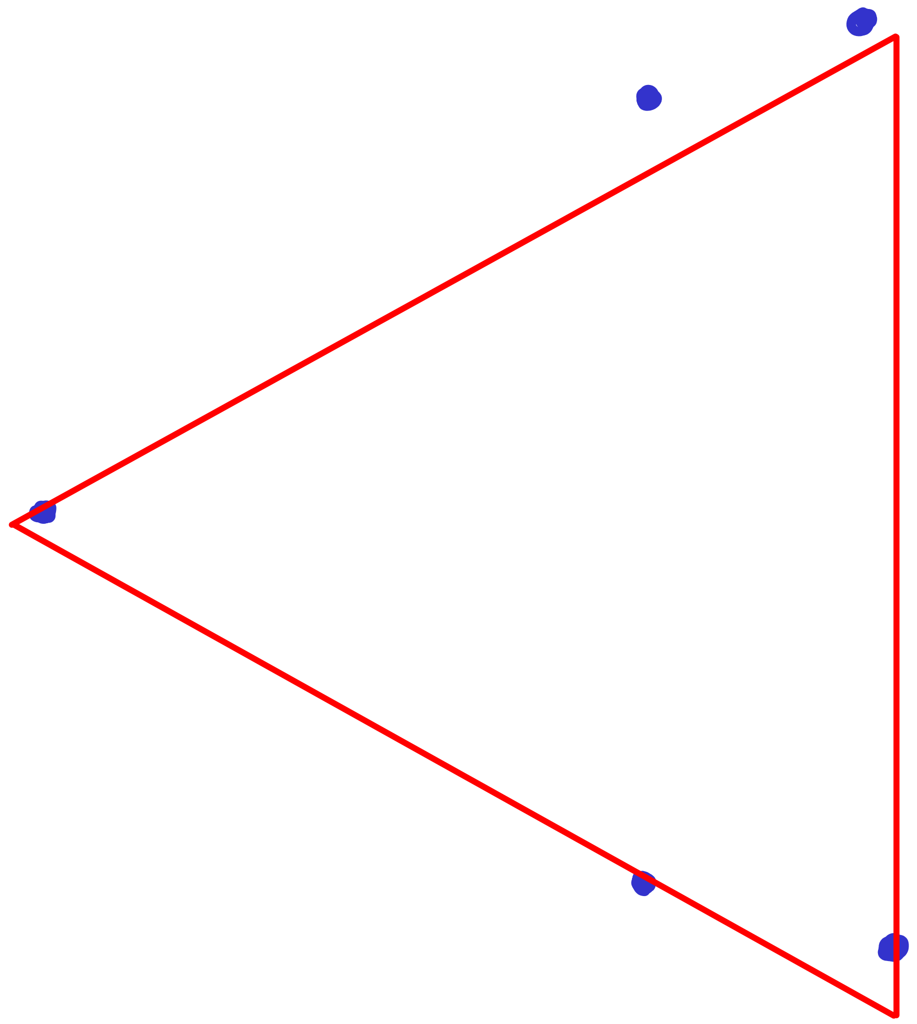


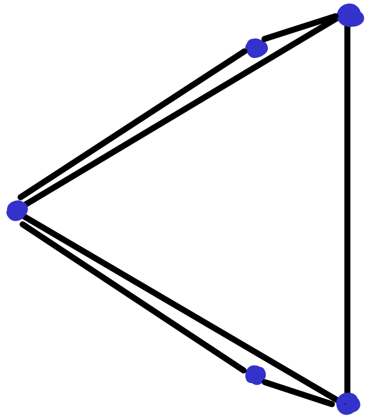
Resulting shape is even simpler

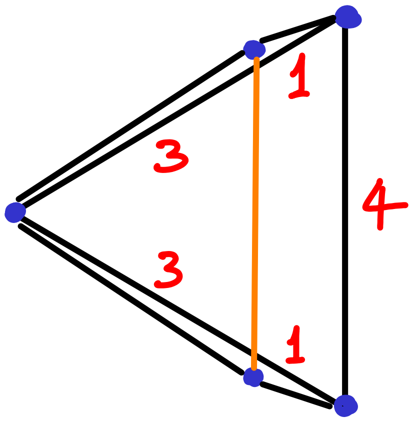








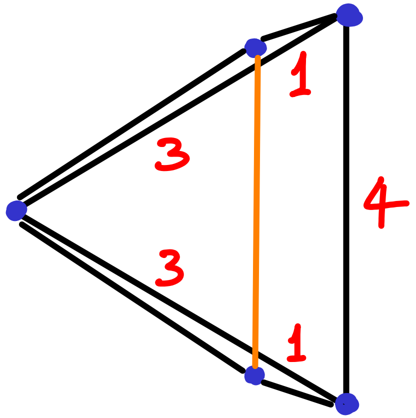




Euclidean $= 3+\varepsilon$

Detour $\simeq 6$

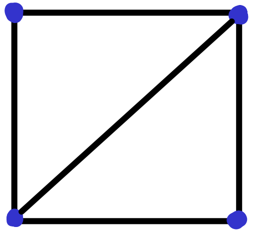
upper bound is tight



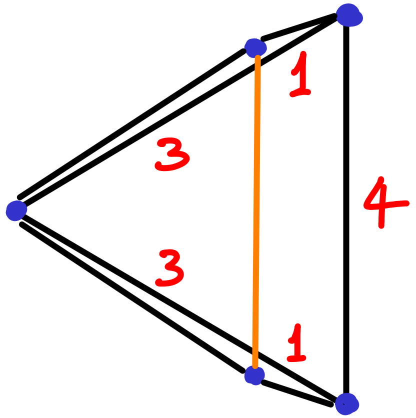
Euclidean $= 3+\varepsilon$

Detour $\simeq 6$

upper bound is tight



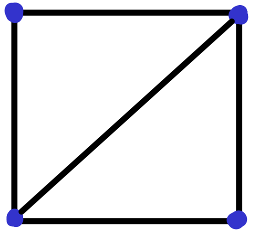
Simple lower bound of $\sqrt{2}$ for any planar spanner



Euclidean $= 3+\varepsilon$

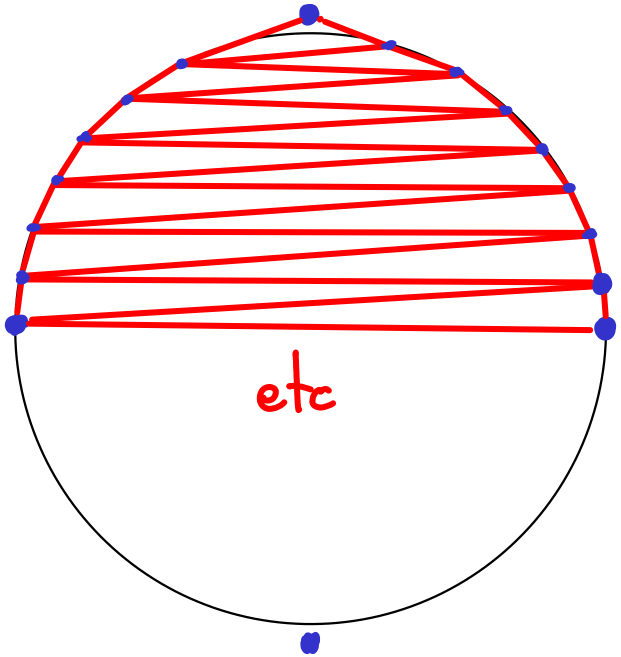
Detour $\simeq 6$

} upper bound is tight

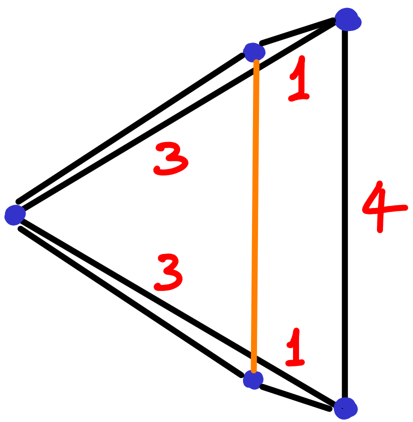


Simple lower bound of $\sqrt{2}$ for any planar spanner

T_{L_2}



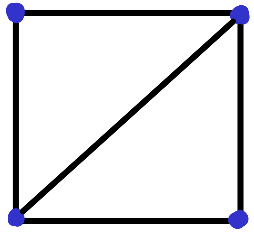
perturbed co-circular points: can choose any Delaunay triangulation



Euclidean $= 3+\varepsilon$

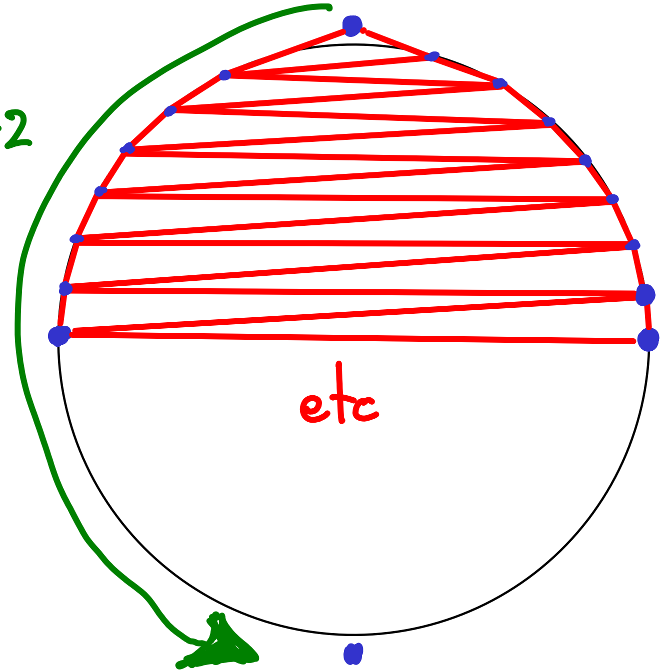
Detour $\simeq 6$

} upper bound is tight



Simple lower bound of $\sqrt{2}$ for any planar spanner

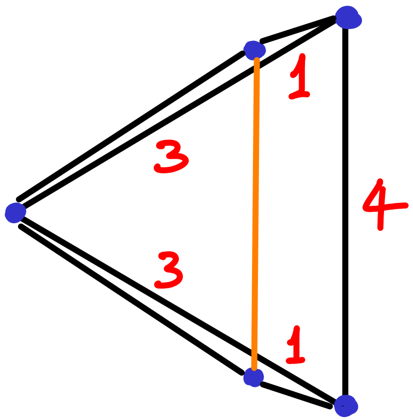
T_{L_2}



perturbed co-circular points: can choose any Delaunay triangulation

Ratio $\geqslant \frac{\pi}{2}$

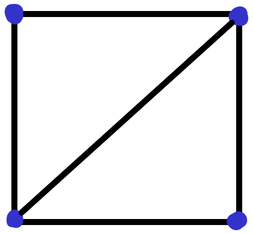
& known $\leqslant \dfrac{2\pi}{3\cos\frac{\pi}{6}} \sim 2.42$



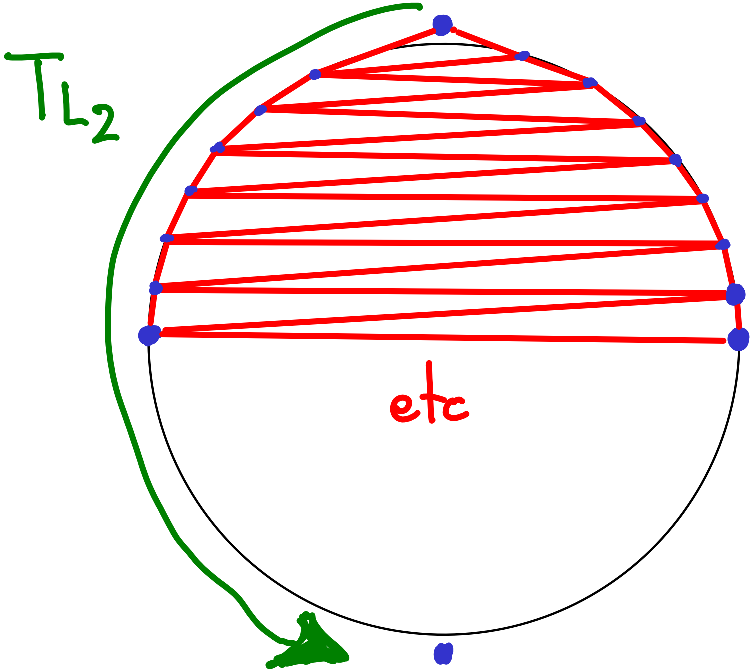
Euclidean $= 3+\varepsilon$

Detour $\simeq 6$

upper bound is tight



Simple lower bound of $\sqrt{2}$ for any planar spanner



perturbed co-circular points: can choose any Delaunay triangulation

Ratio $> \frac{\pi}{2}$

improved by Bose et al.

& known $\leq \frac{2\pi}{3\cos\frac{\pi}{6}} \sim 2.42$