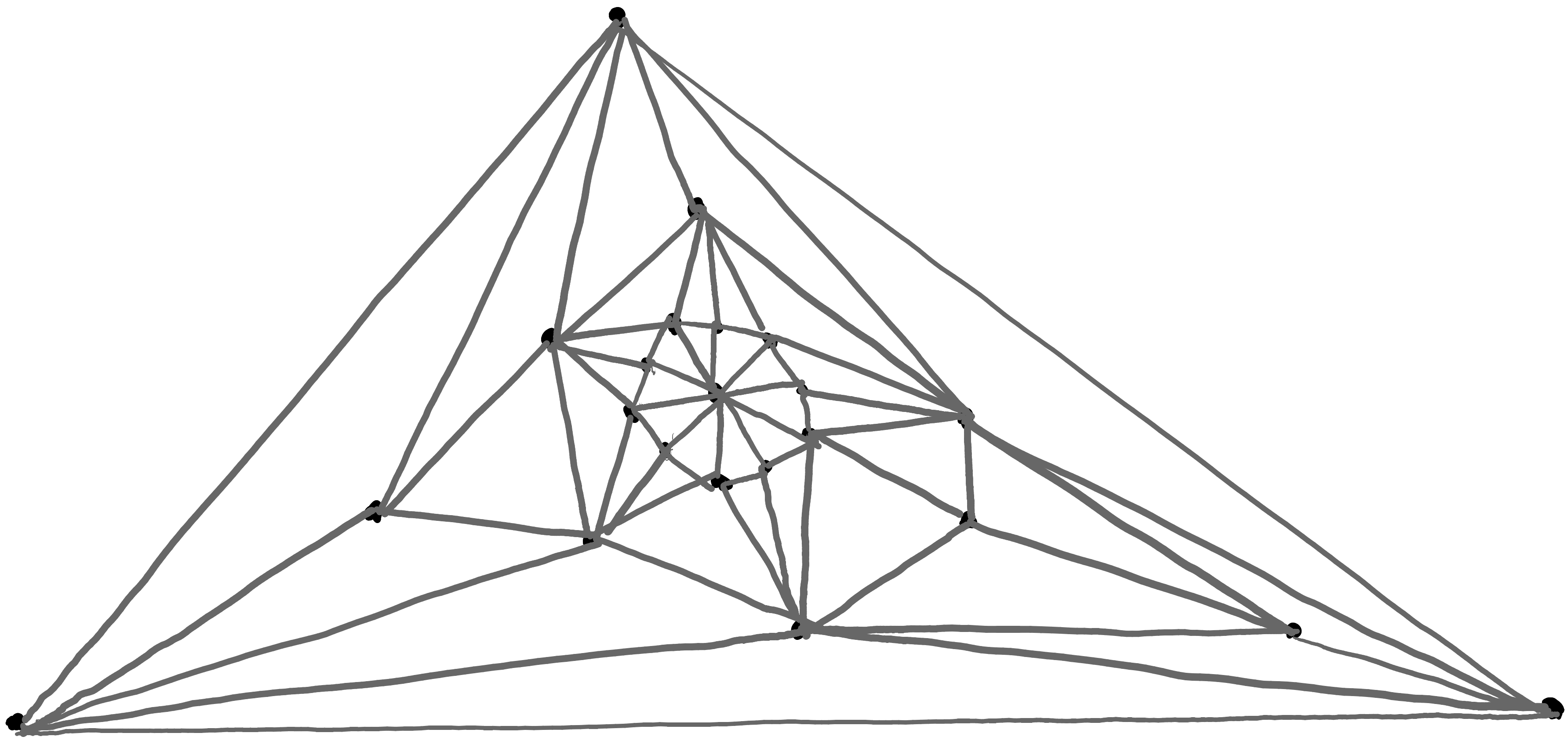
Kirkpatrick's Method

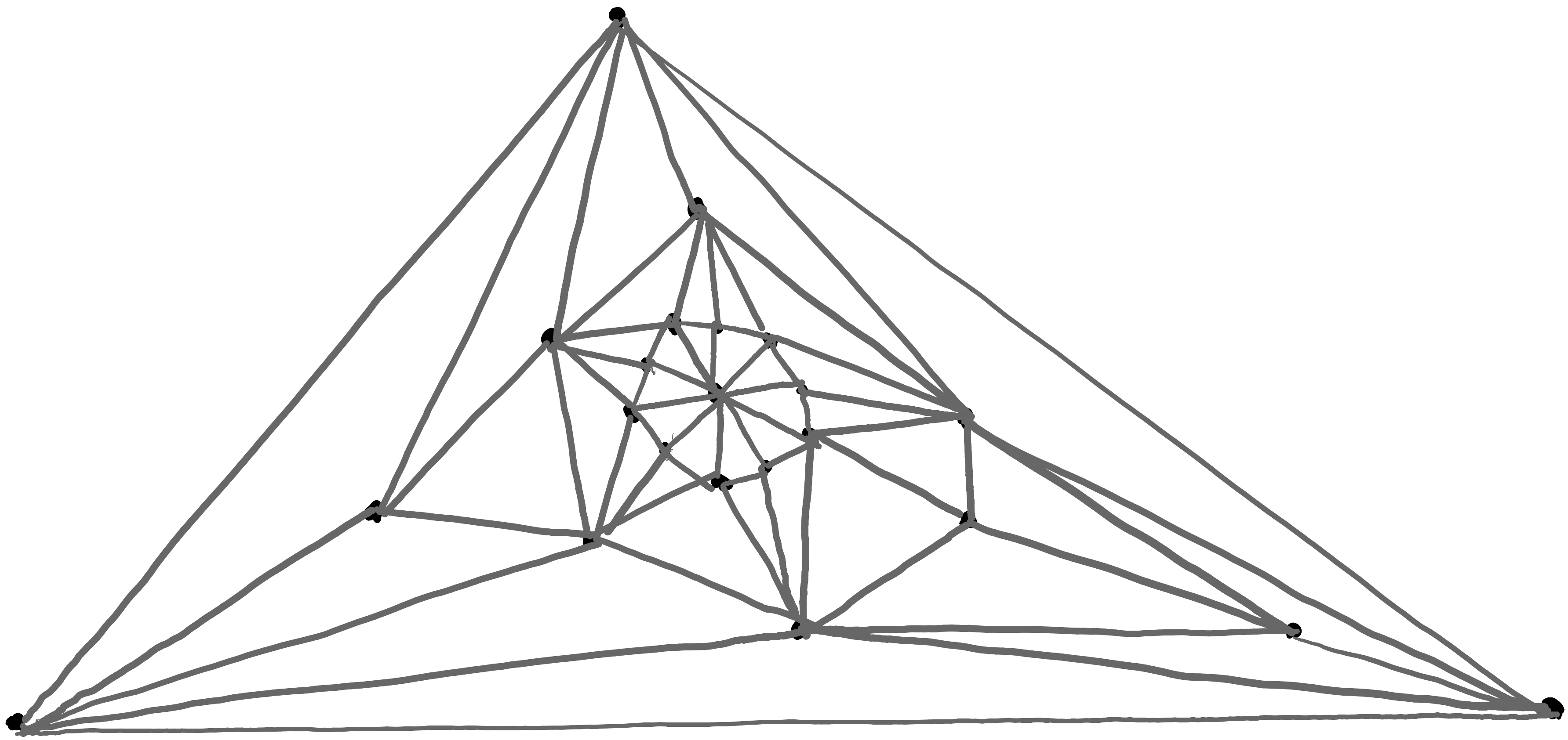
SIAM J. Comput 12(1) 1983

John Iacono

The Problem



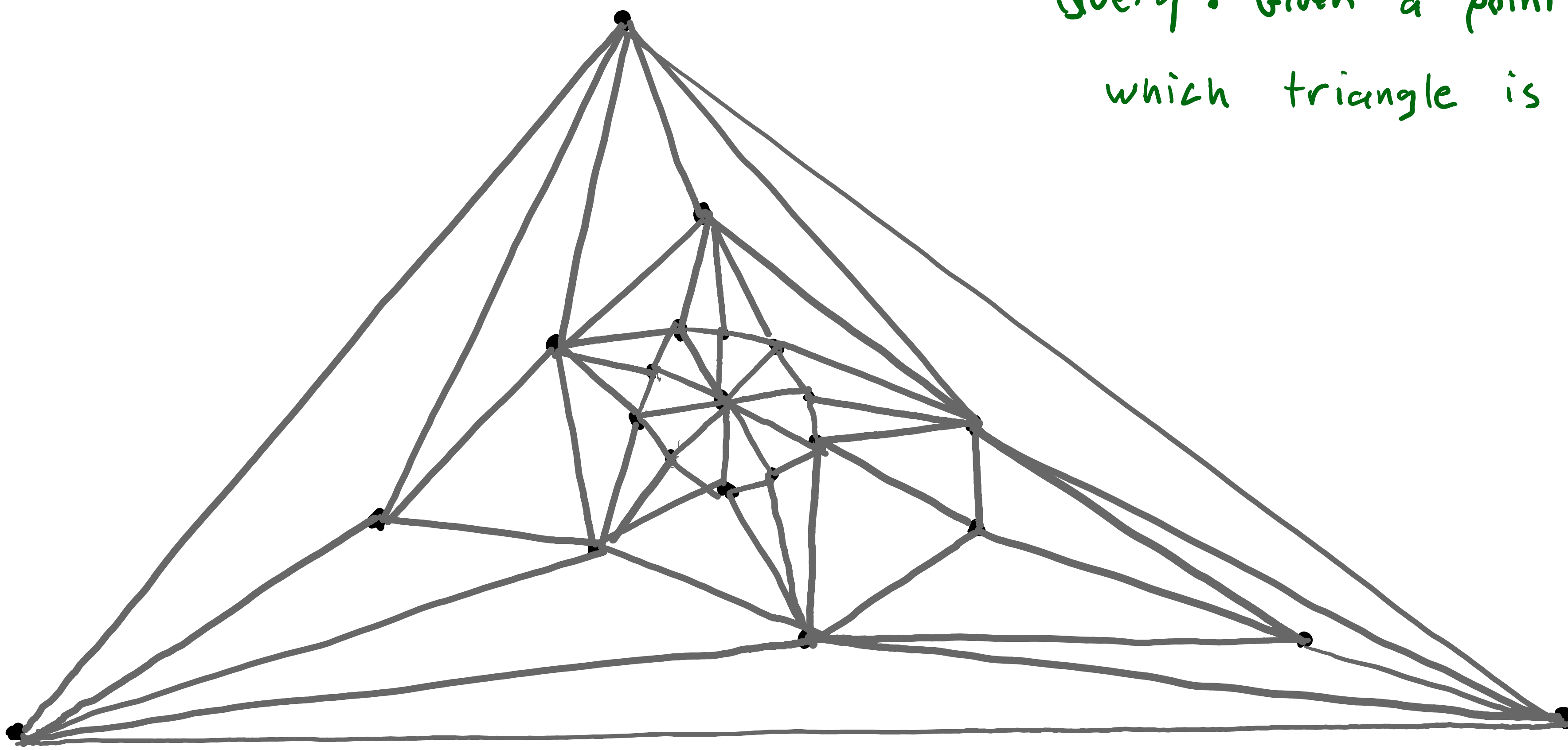
Input: A triangulation



Input: A triangulation

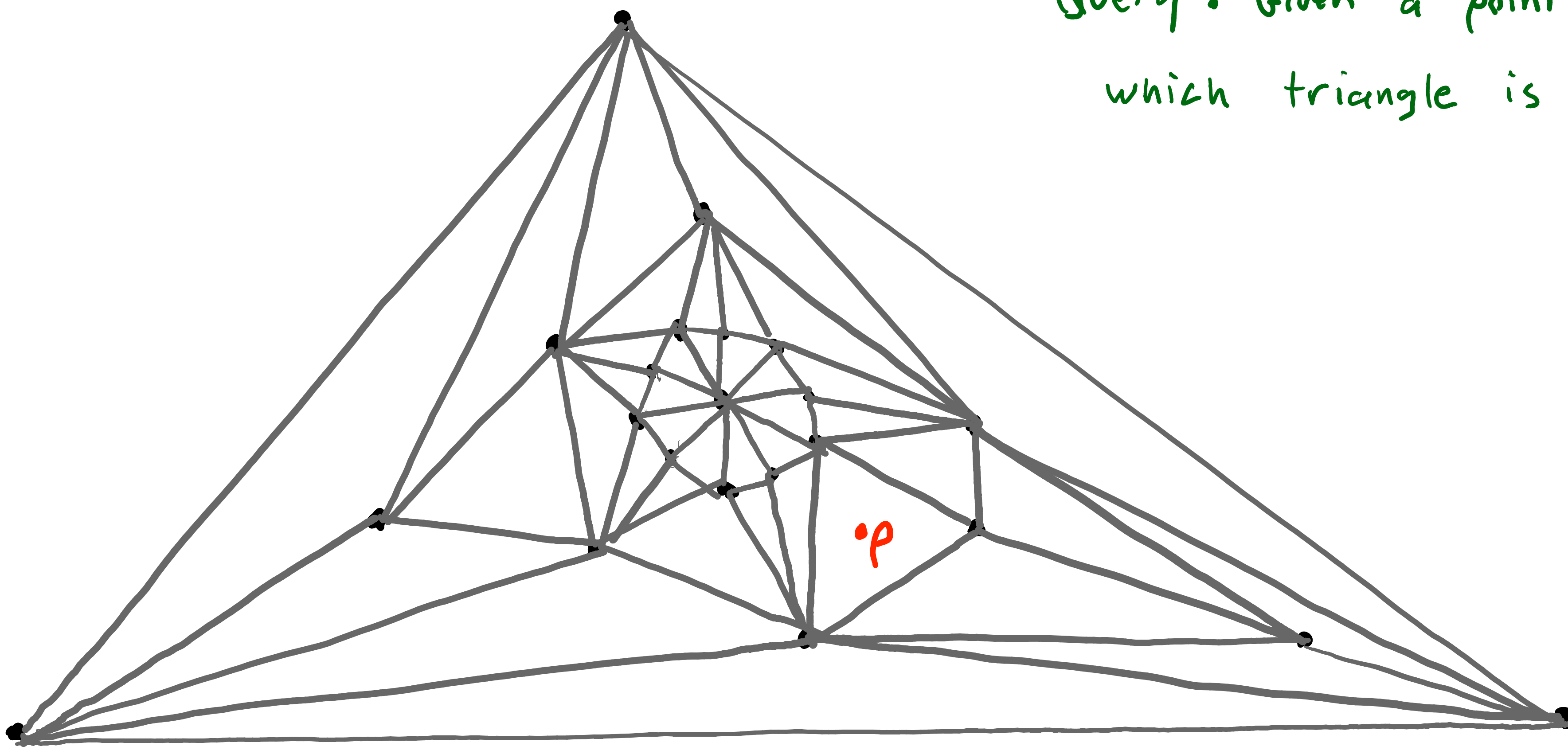
Query: Given a point p,

which triangle is p in?



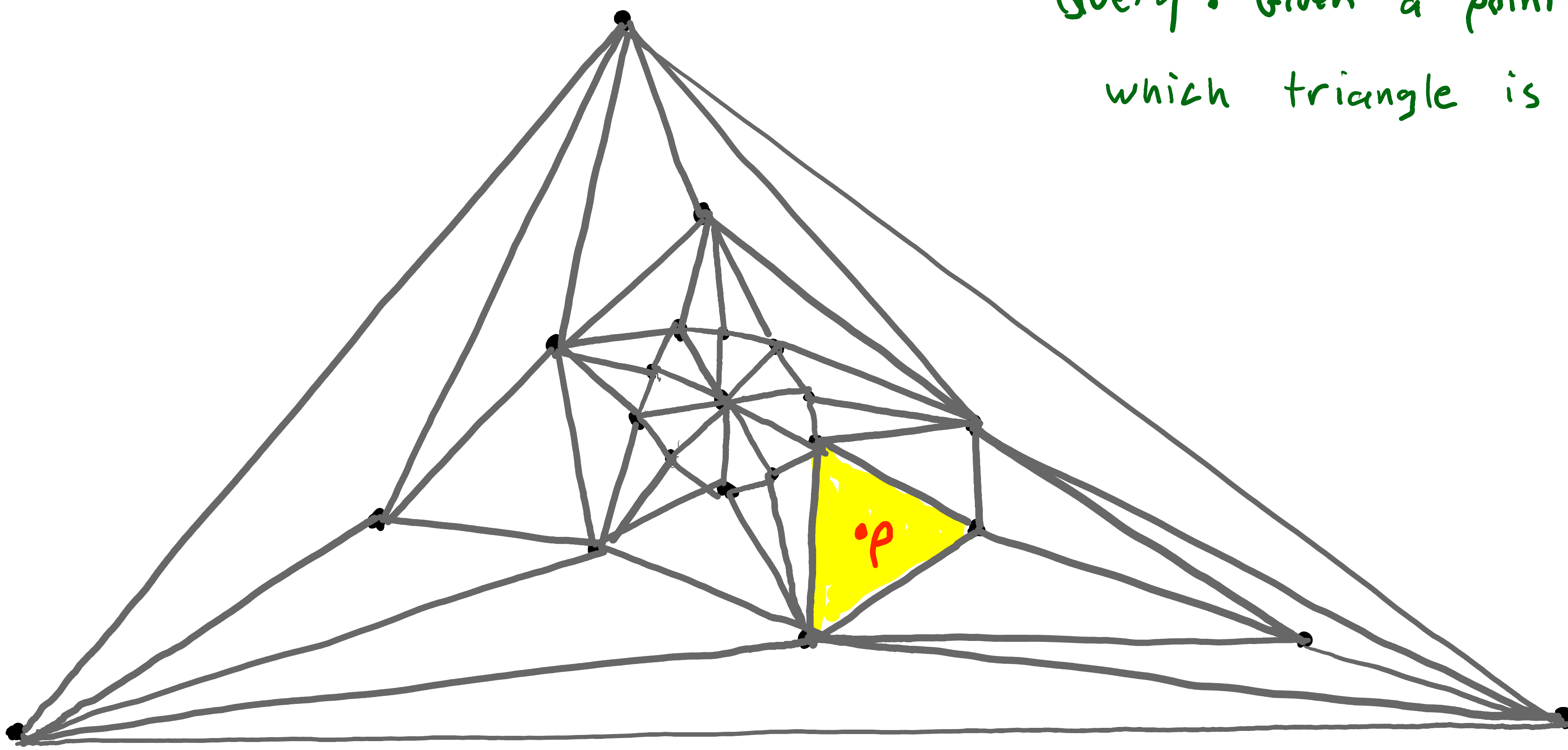
Input: A triangutation

Query: Given a point p, which triangle is p in?



Input: A triangulation

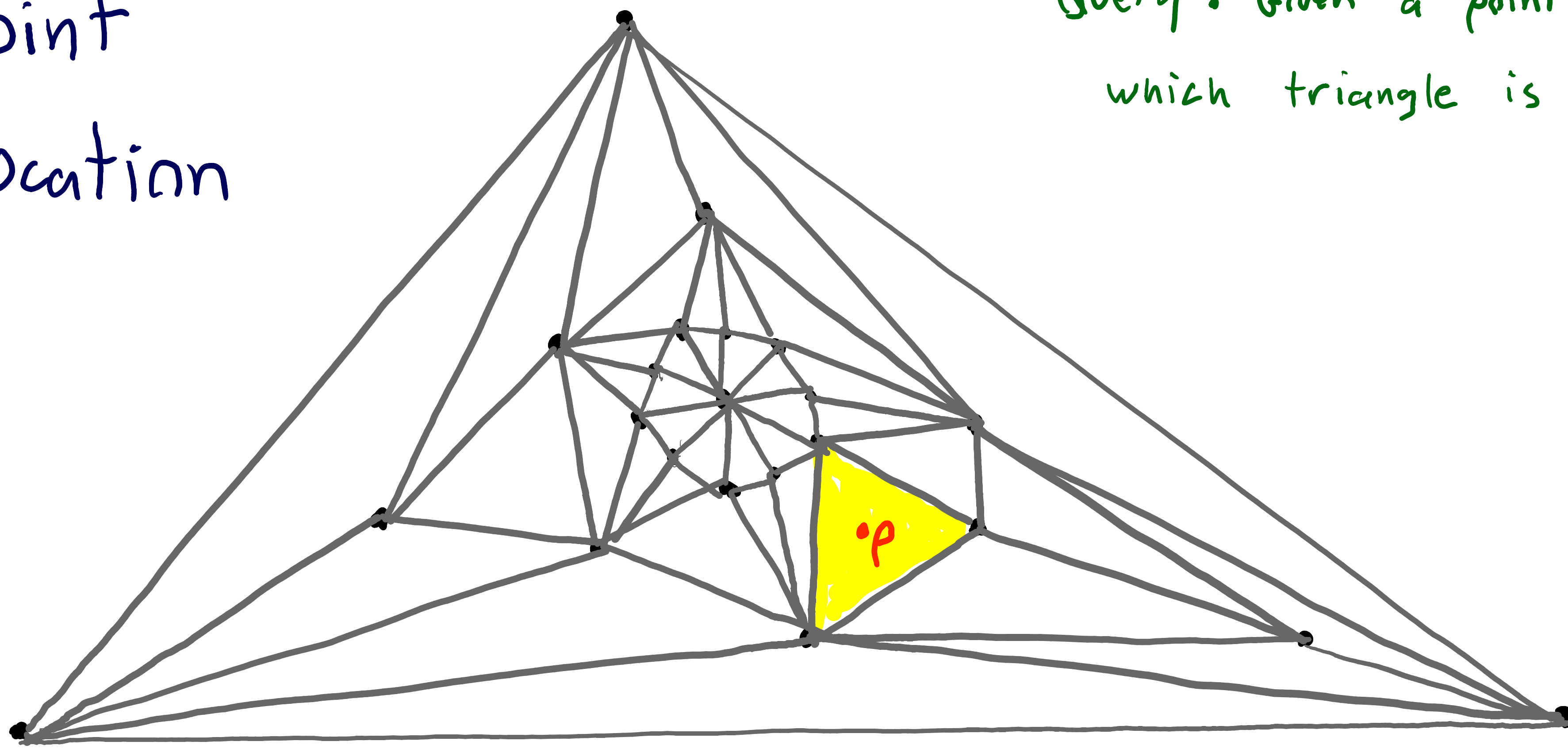
Query: Given a point p, which triangle is p in?



Planar Point Location

Input: A triangulation

Query: Given a point p, which triangle is p in?



Basic Idea: Refinement

Basic Idea: Refinement

Let's look at 1-D

Basic Idea: Refinement

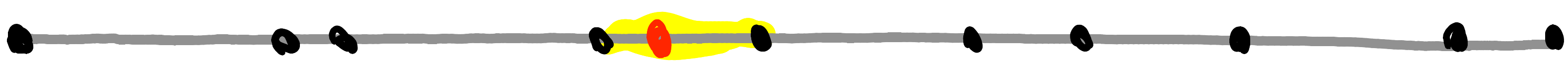
Let's look at 1-D

Basic Idea: Refinement

Let`s look at 1-D

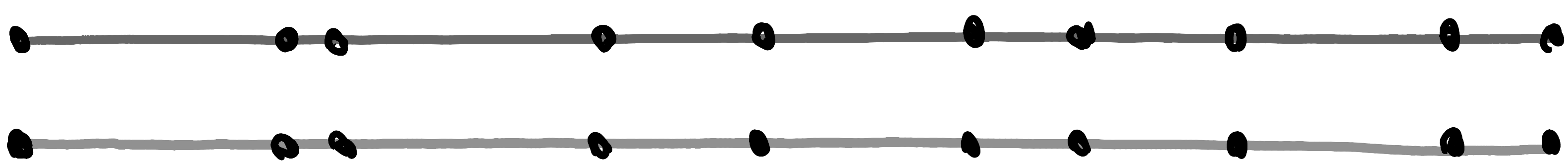
Basic Idea: Refinement

Let's look at 1-D



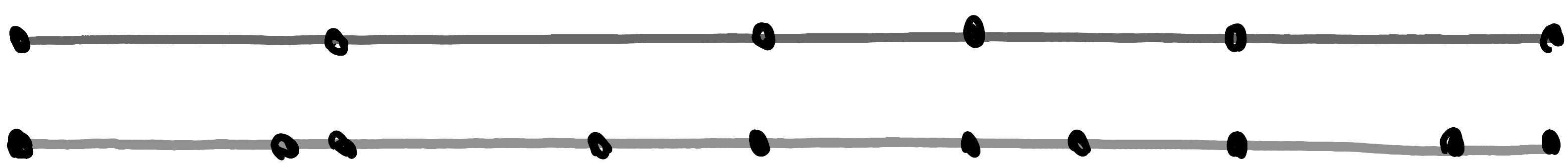
Basic Idea: Refinement

Let's look at 1-D



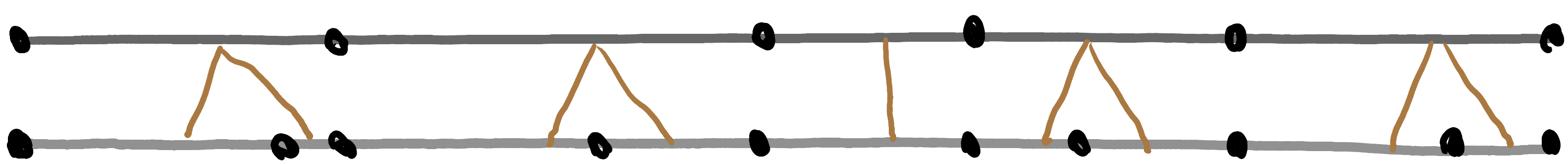
Basic Idea: Refinement

Let's look at 1-D



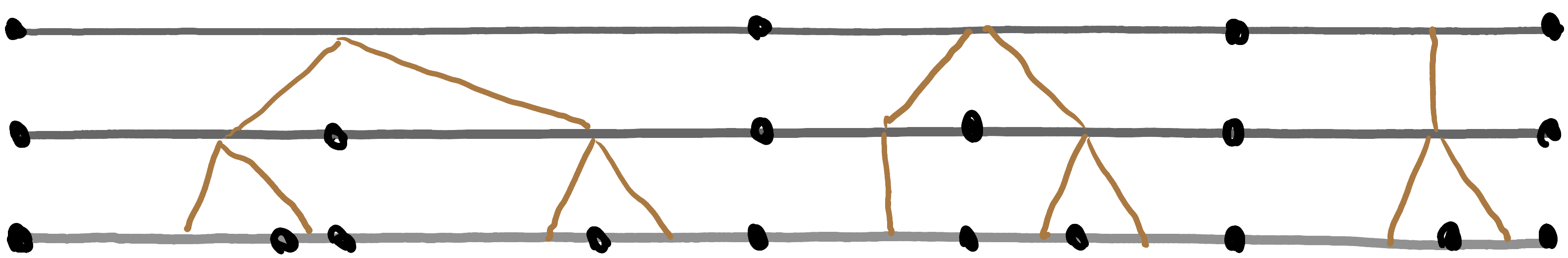
Basic Idea: Refinement

Let's look at 1-D



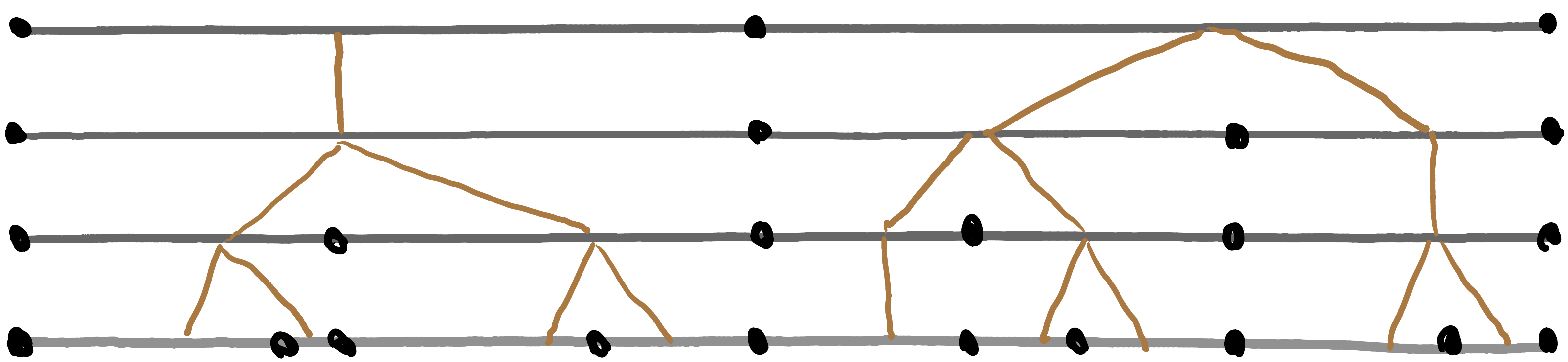
Basic Idea: Refinement

Let's look at 1-D



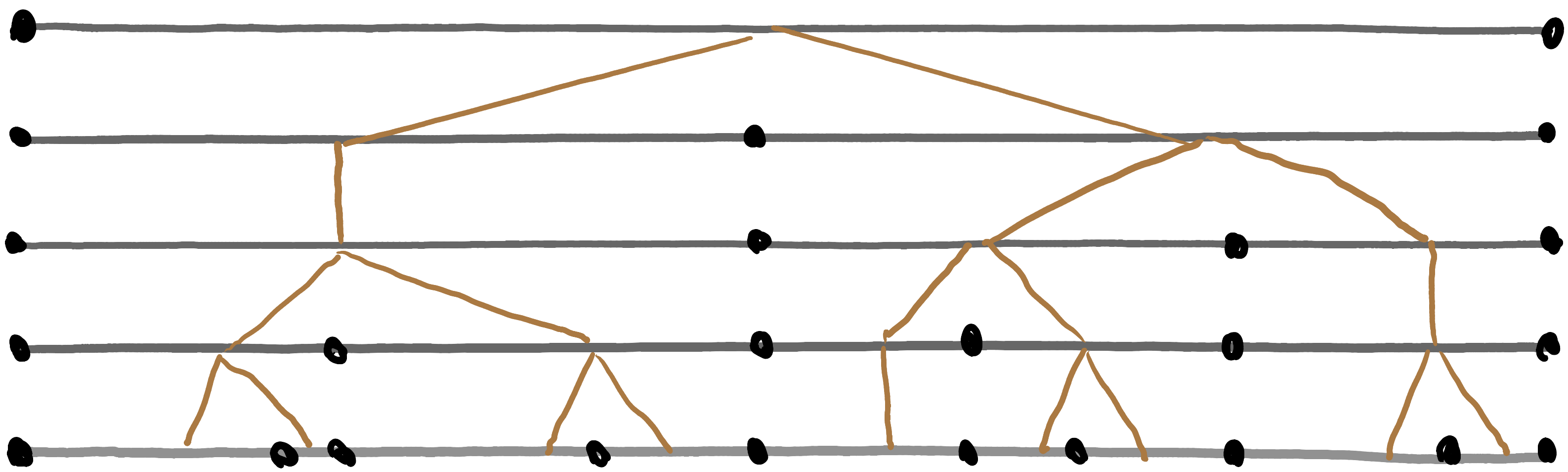
Basic Idea: Refinement

Let's look at 1-D



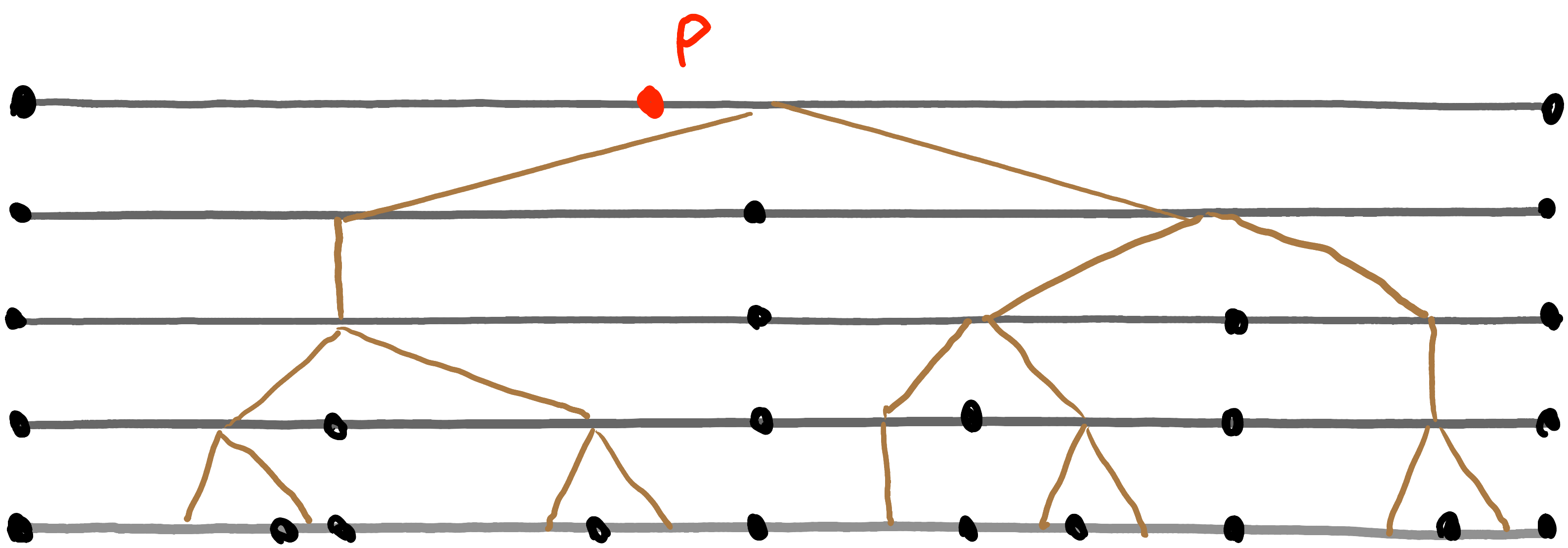
Basic Idea: Refinement

Let's look at 1-D



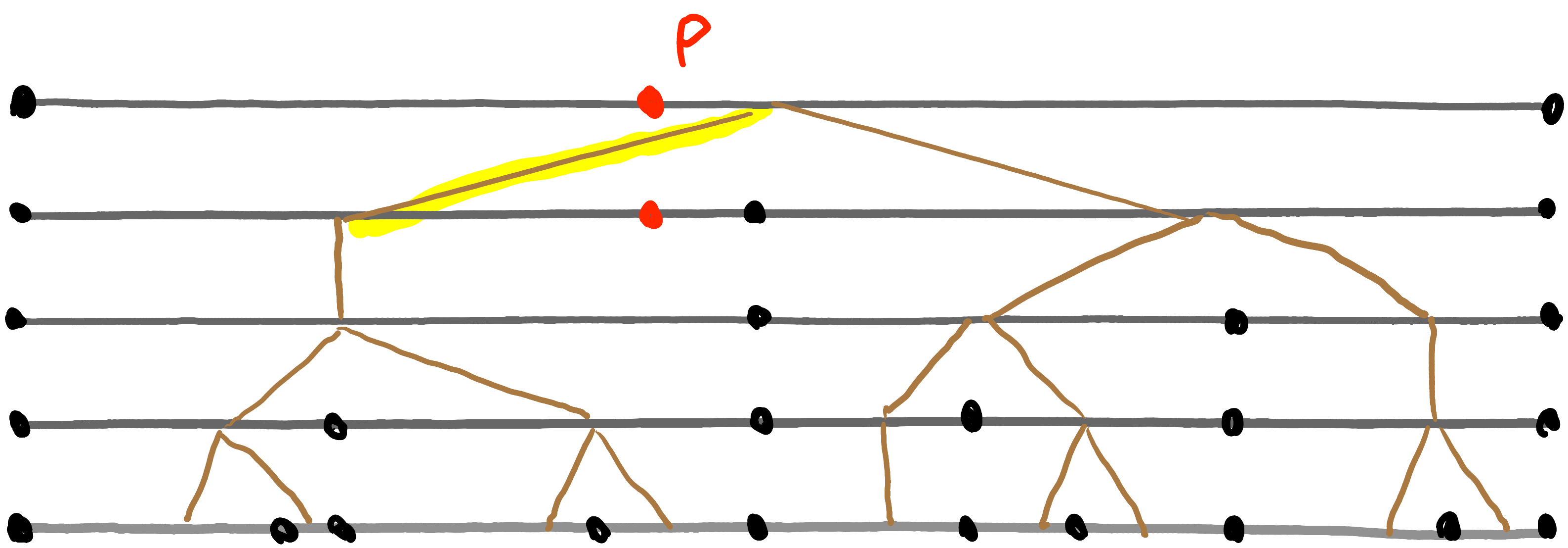
Basic Idea: Refinement

Let's look at 1-D



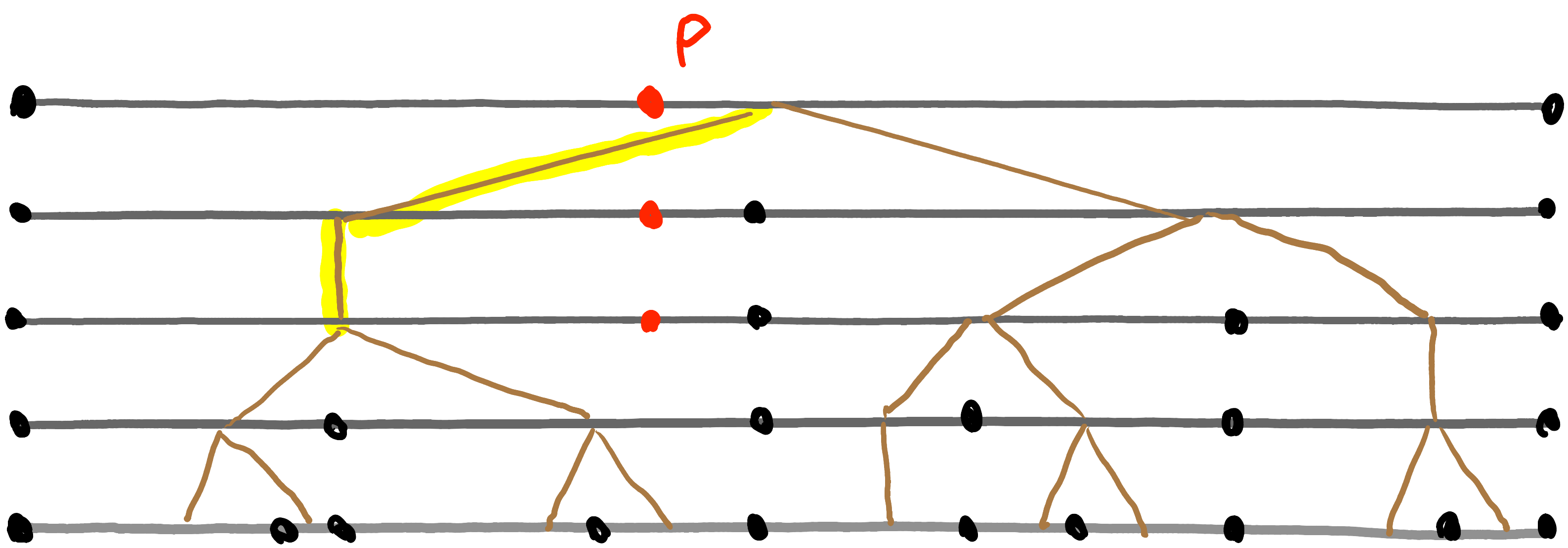
Basic Idea: Refinement

Let's look at 1-D



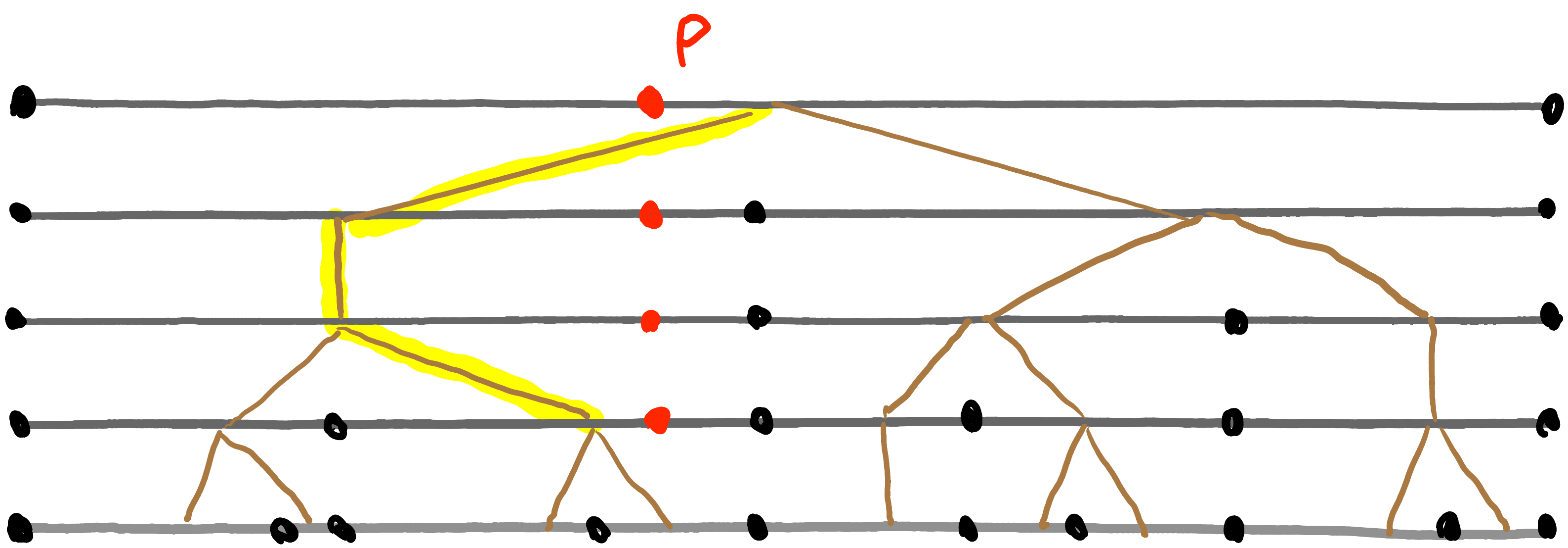
Basic Idea: Refinement

Let's look at 1-D



Basic Idea: Refinement

Let's look at 1-D



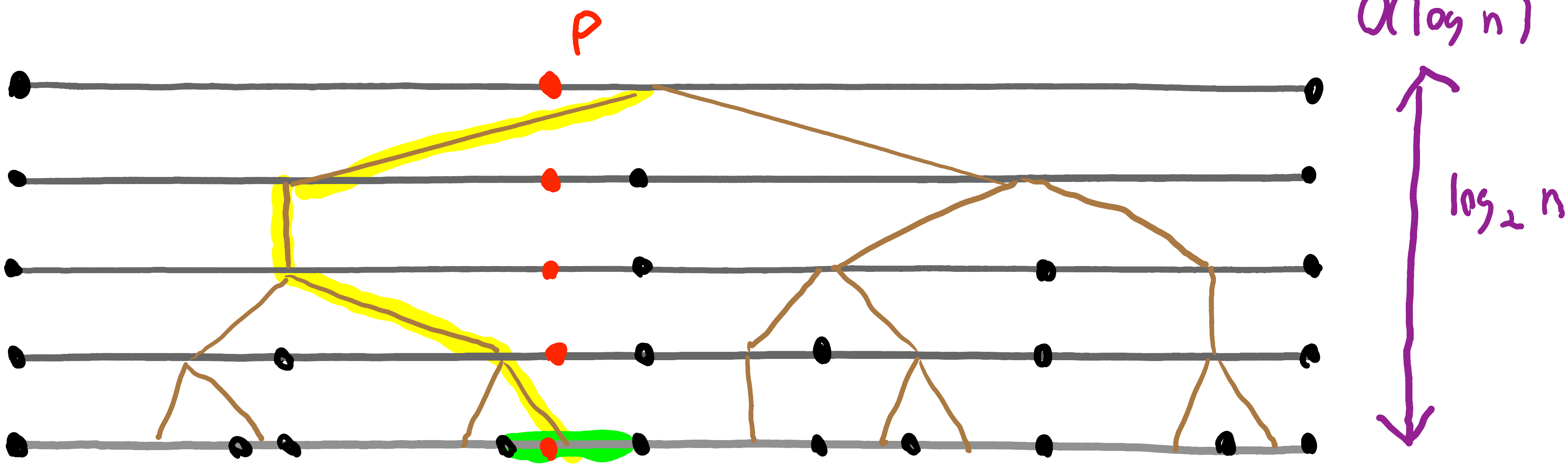
Analysis

Construction time: $O(n)$

$T(n) = T(\frac{n}{2}) + n$

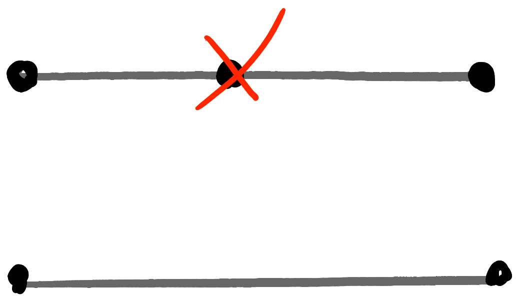
Query

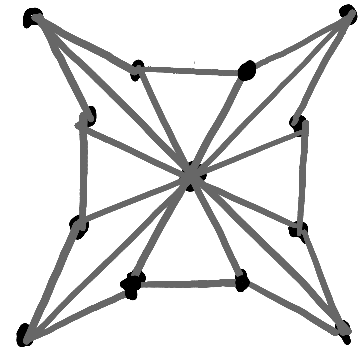
$O(\log n)$



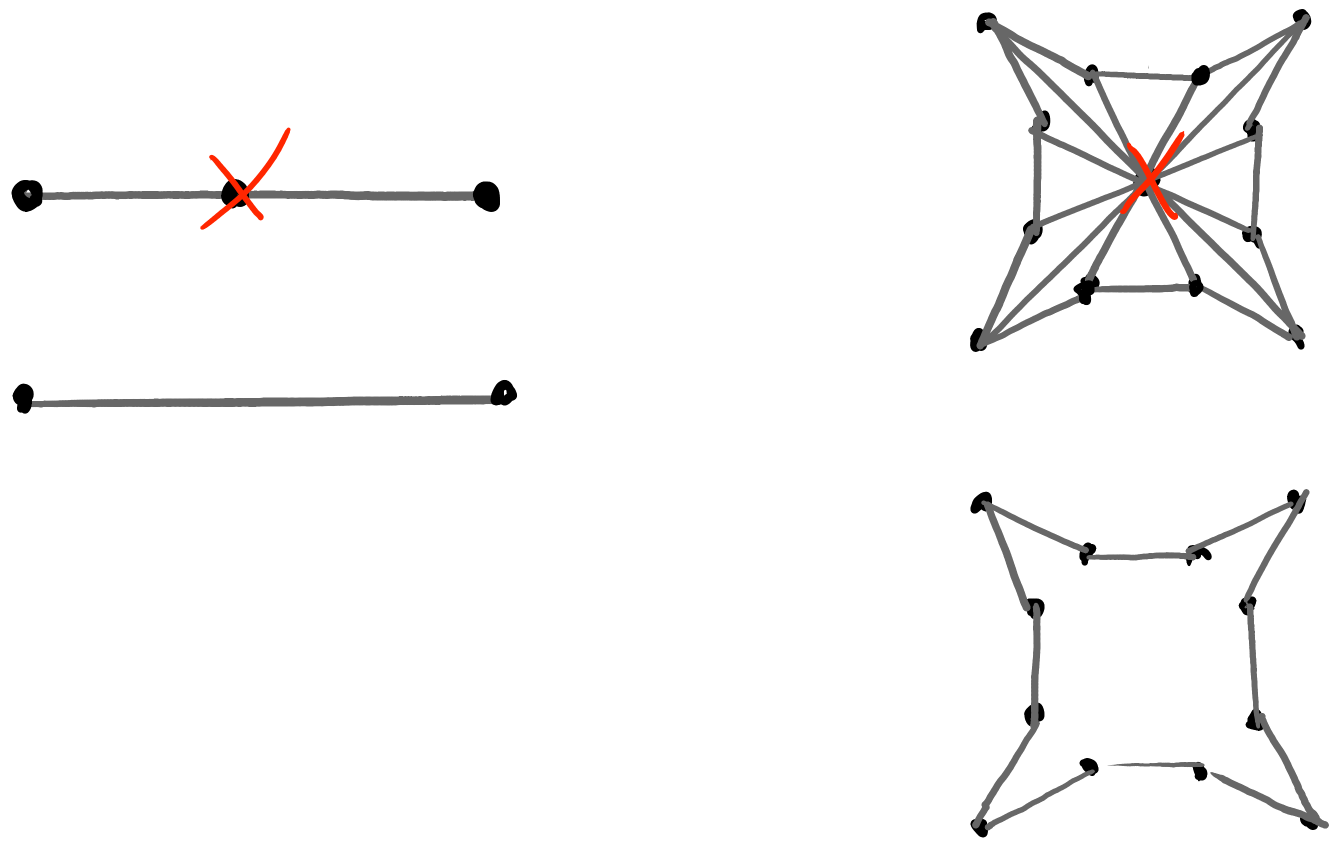
1D $\rightsquigarrow$ 2D

1D ⟶ 2D

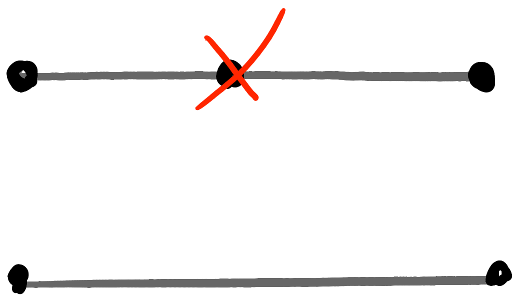


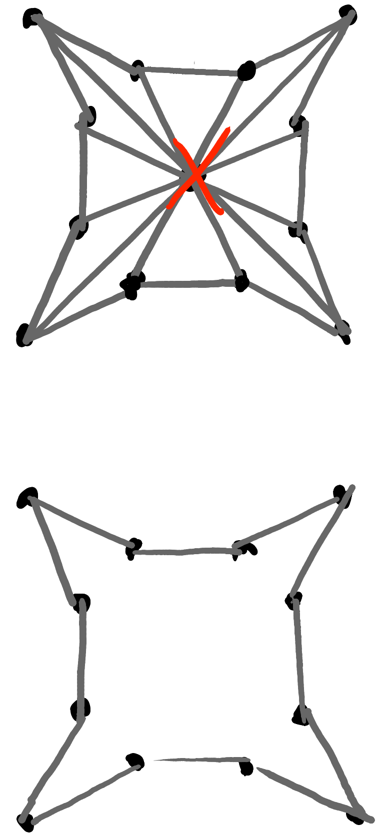


1D ⟶ 2D



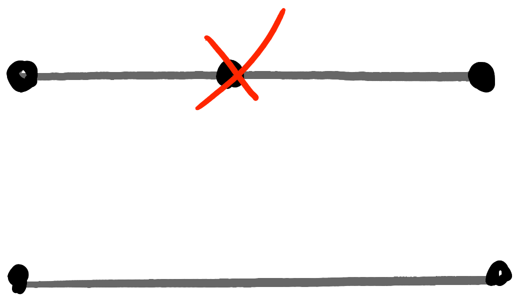
1D ⟶ 2D

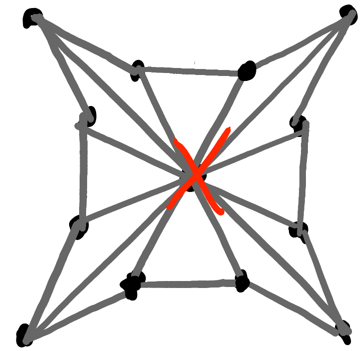




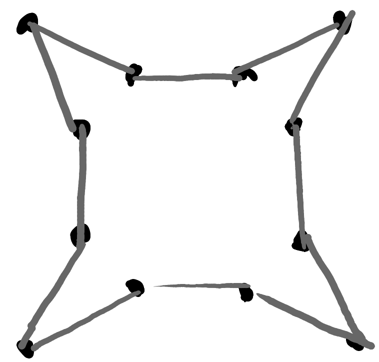
- Not a triangulation
- Possibly linear size

1D $\longrightarrow$ 2D



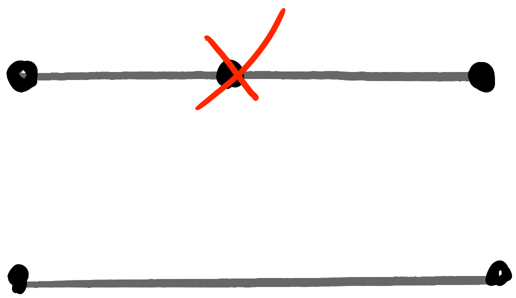


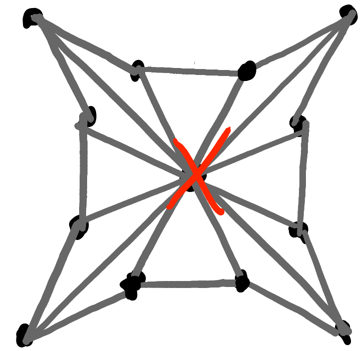
- Every polygon can be triangulated (will prove later)



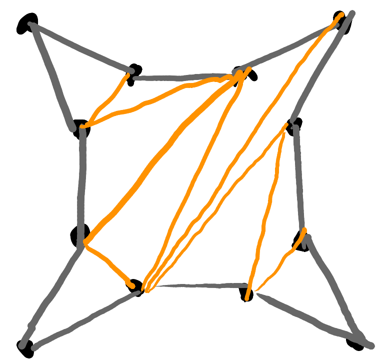
- Not a triangulation
- Possibly linear size

1D → 2D



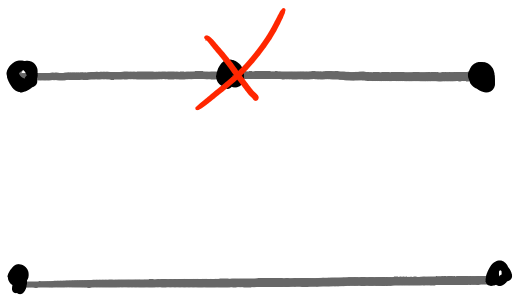


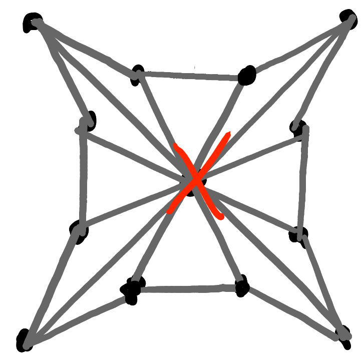
- Every polygon can be triangulated (will prove later)

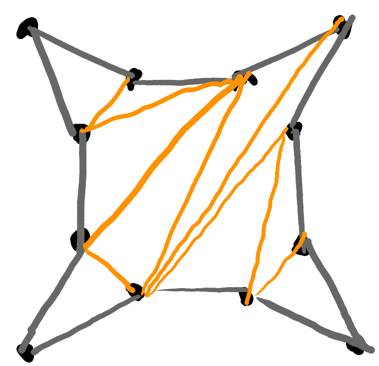


- Not a triangulation
- Possibly linear size

1D ⟶ 2D







- Not a triangulation
- Possibly linear size

- Every polygon can be triangulated (will prove later)

- Most vertexes in a triangulation have small degree (will prove now)

Most vertexes in a triangulation have constant degree

- If an average American weighs 200 lbs, then half of all Americans weigh less than 400 lbs.

Most vertexes in a triangulation have constant degree

- If an average American weighs 200 lbs, then half of all Americans weigh less than 400 lbs.
- From Euler's formula average degree of planar graph is < 6; Thus half of the vertexes have degree < 12

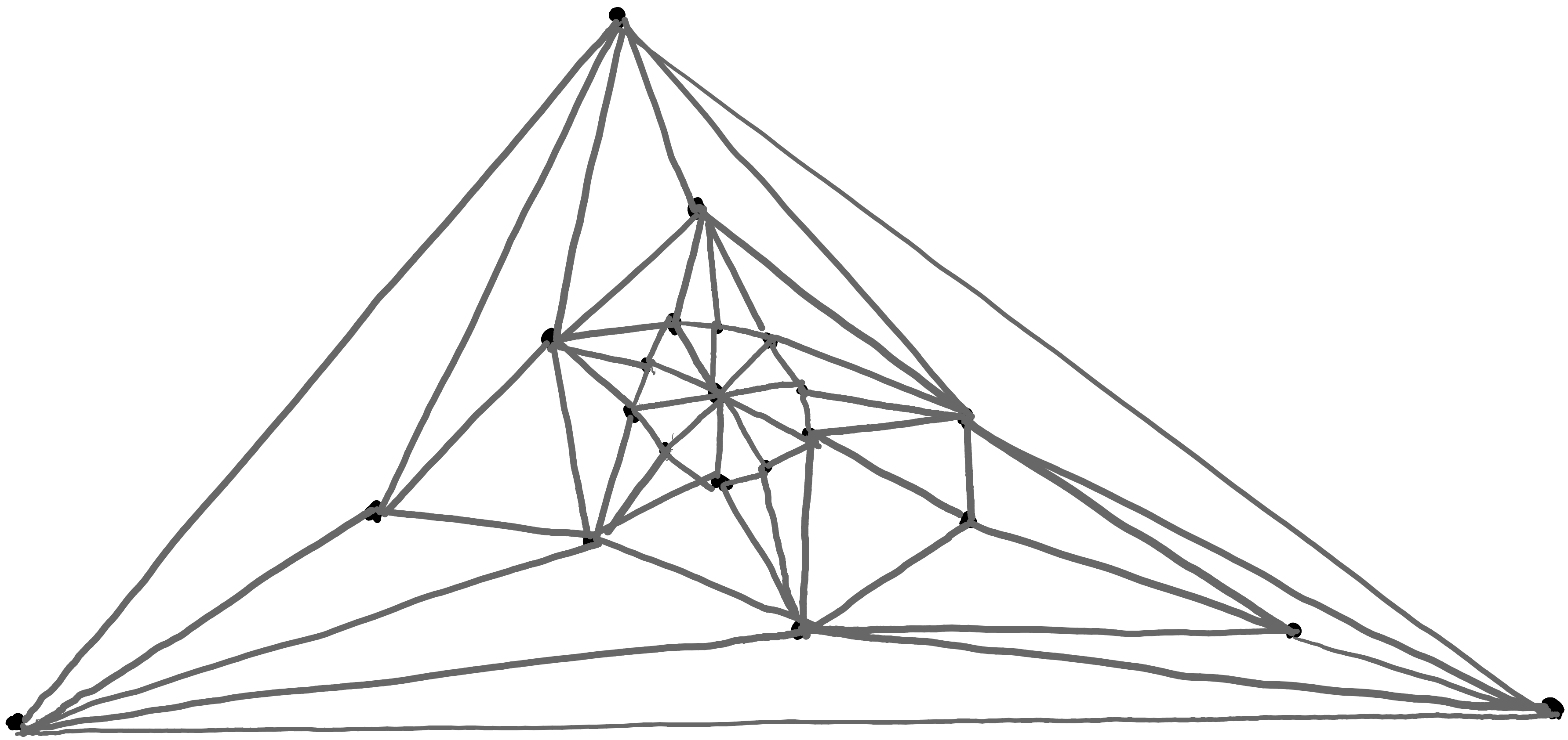
Most vertexes in a triangulation have constant degree

- If an average American weighs 200 lbs, then half of all Americans weigh less than 400 lbs.
- From Euler's formula average degree of planar graph is < 6; Thus half of the vertexes have degree < 12
- All interior vertexes of a triangulation have degree ≥ 3
  Thus half of the vertexes have degree < 9

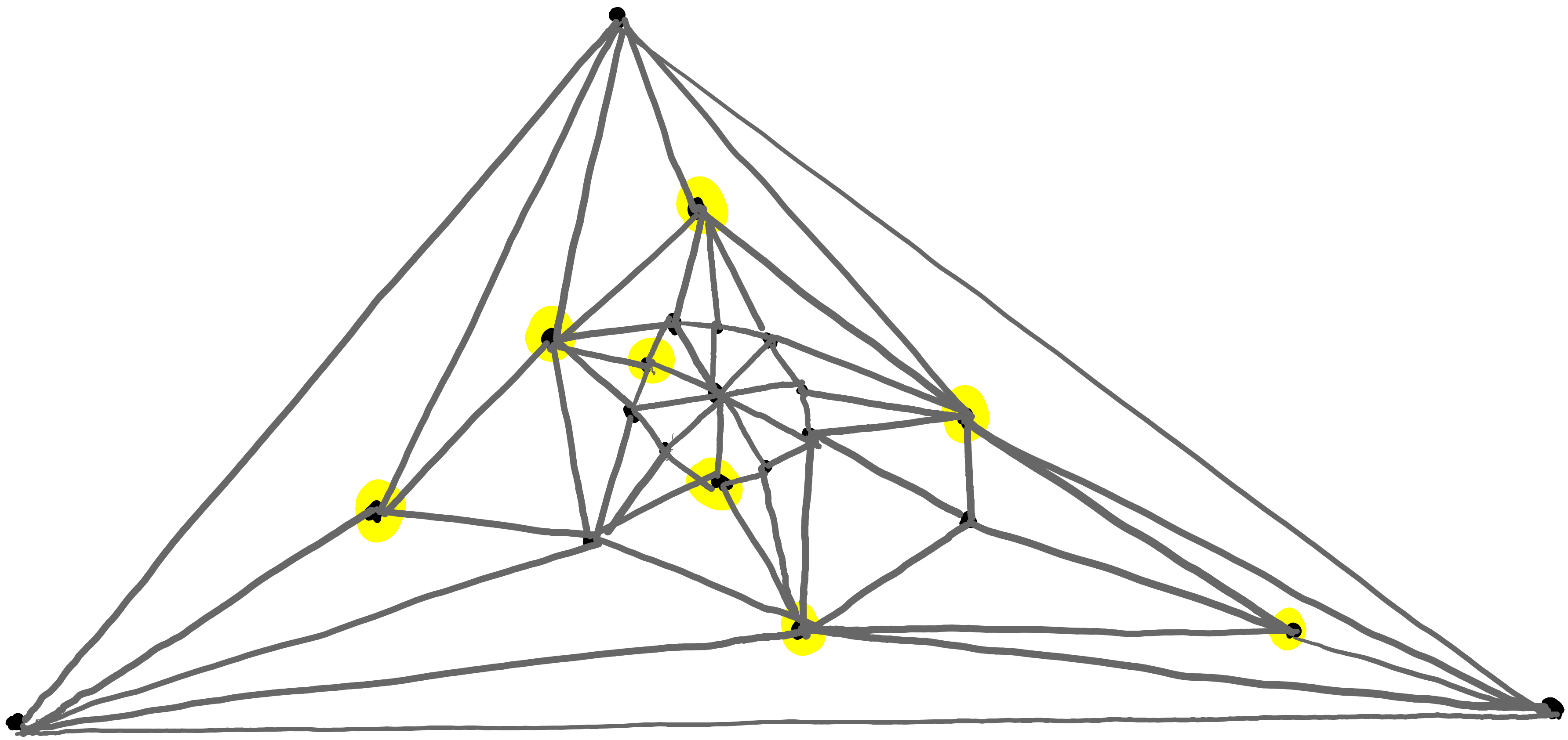
Most vertexes in a triangulation have constant degree

- If an average American weighs 200 lbs, then half of all Americans weigh less than 400 lbs.
- From Euler's formula average degree of planar graph is < 6; Thus half of the vertexes have degree < 12
- All interior vertexes of a triangulation have degree ≥ 3 Thus half of the vertexes have degree < 9
- $\geq$ Half of interior vertexes have degree 3, 4, 5, 6, 7, 8

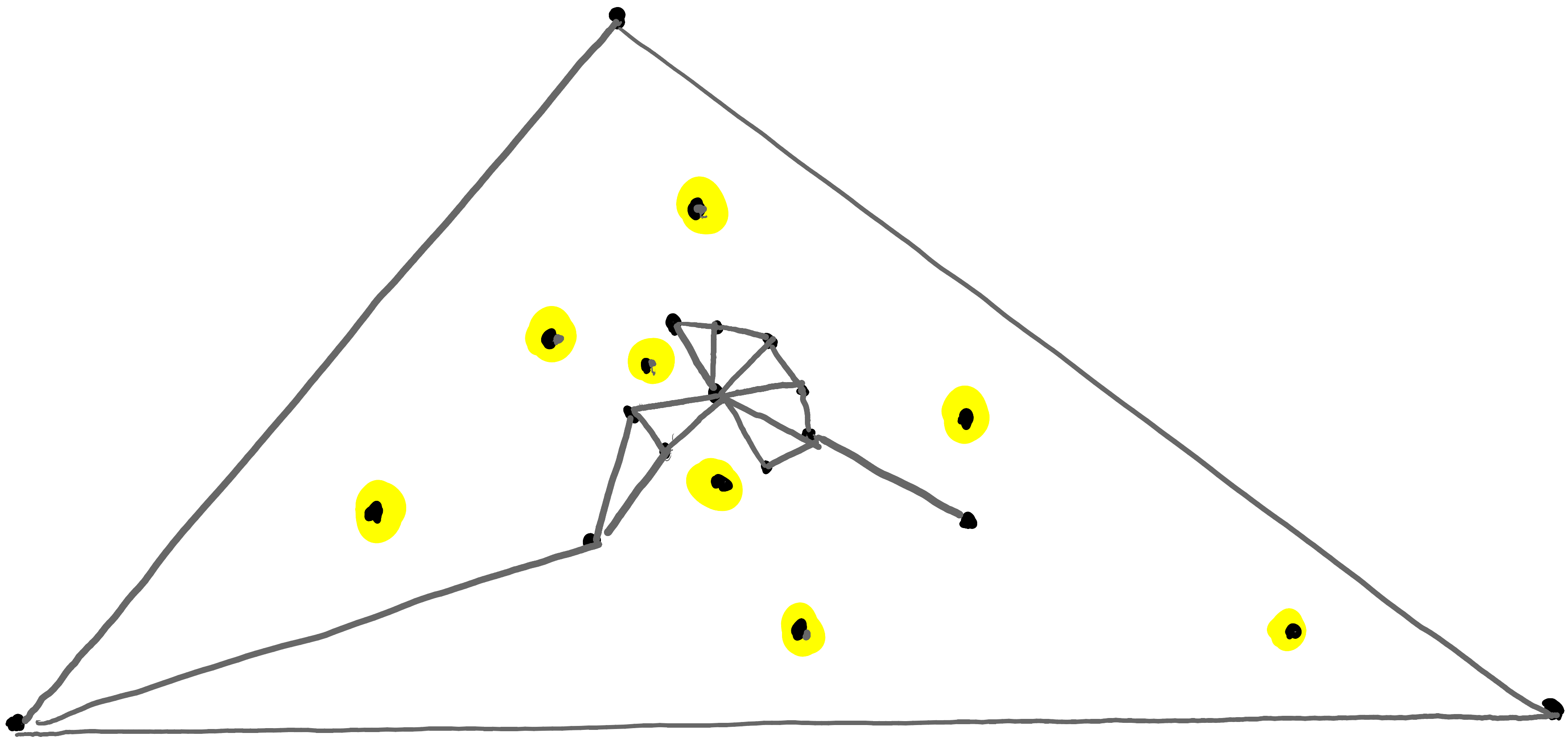
I see 17 interior vertexes, can I remove 8 with degree ≤8?



I see 17 interior vertexes, can I remove 8 with degree ≤8?

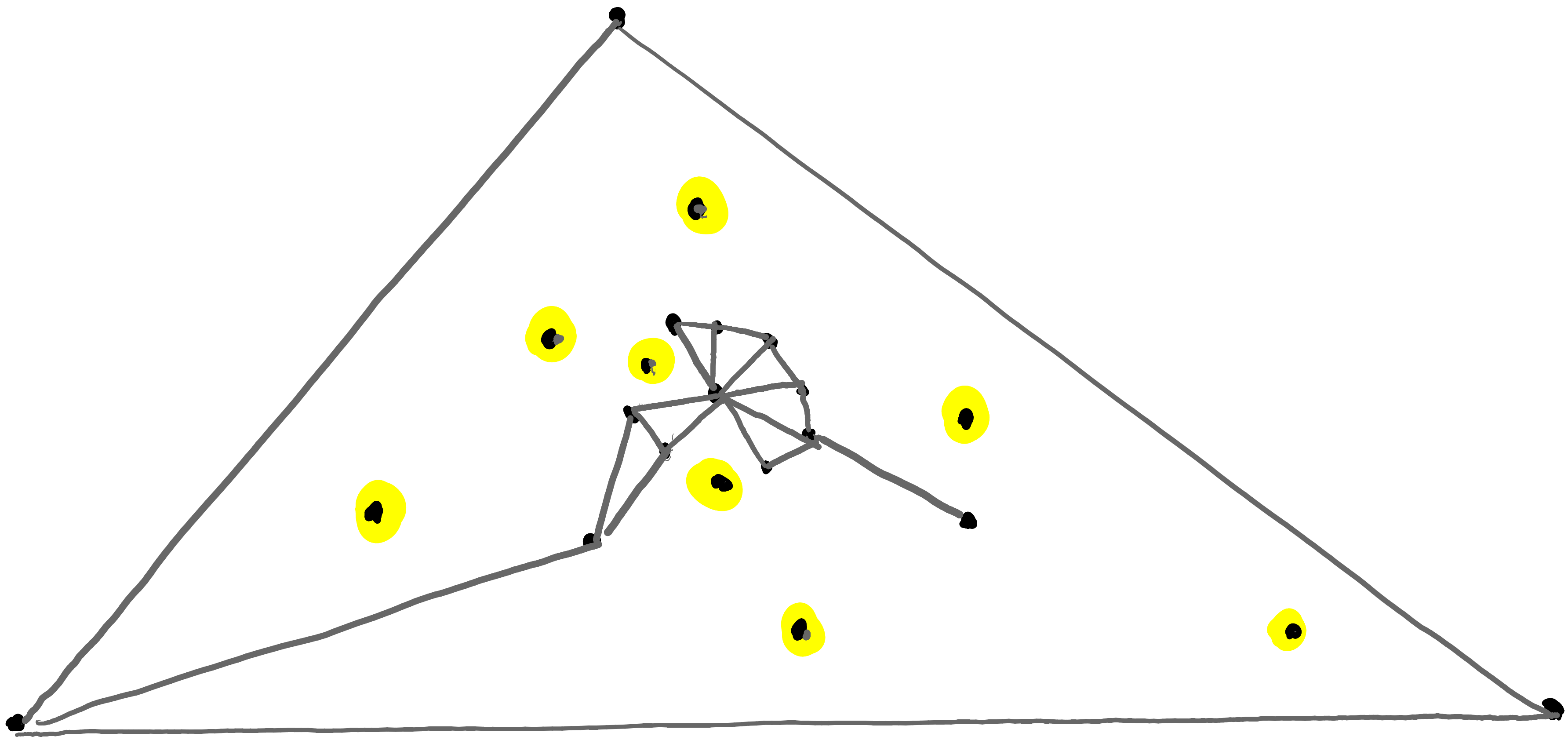


I see 17 interior vertexes, can I remove 8 with degree ≤ 8?

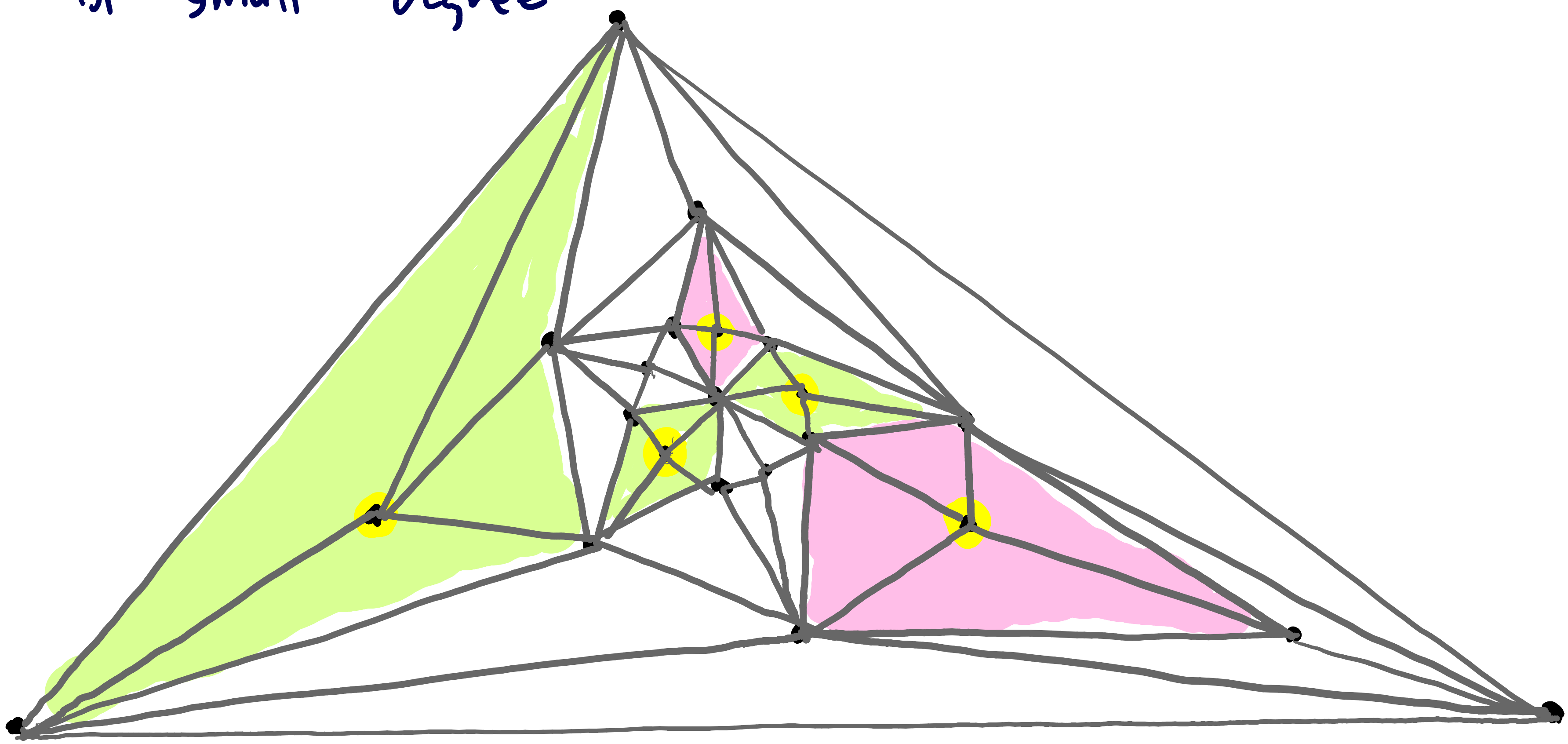


I see 17 interior vertexes, can I remove 8 with degree ≤ 8?

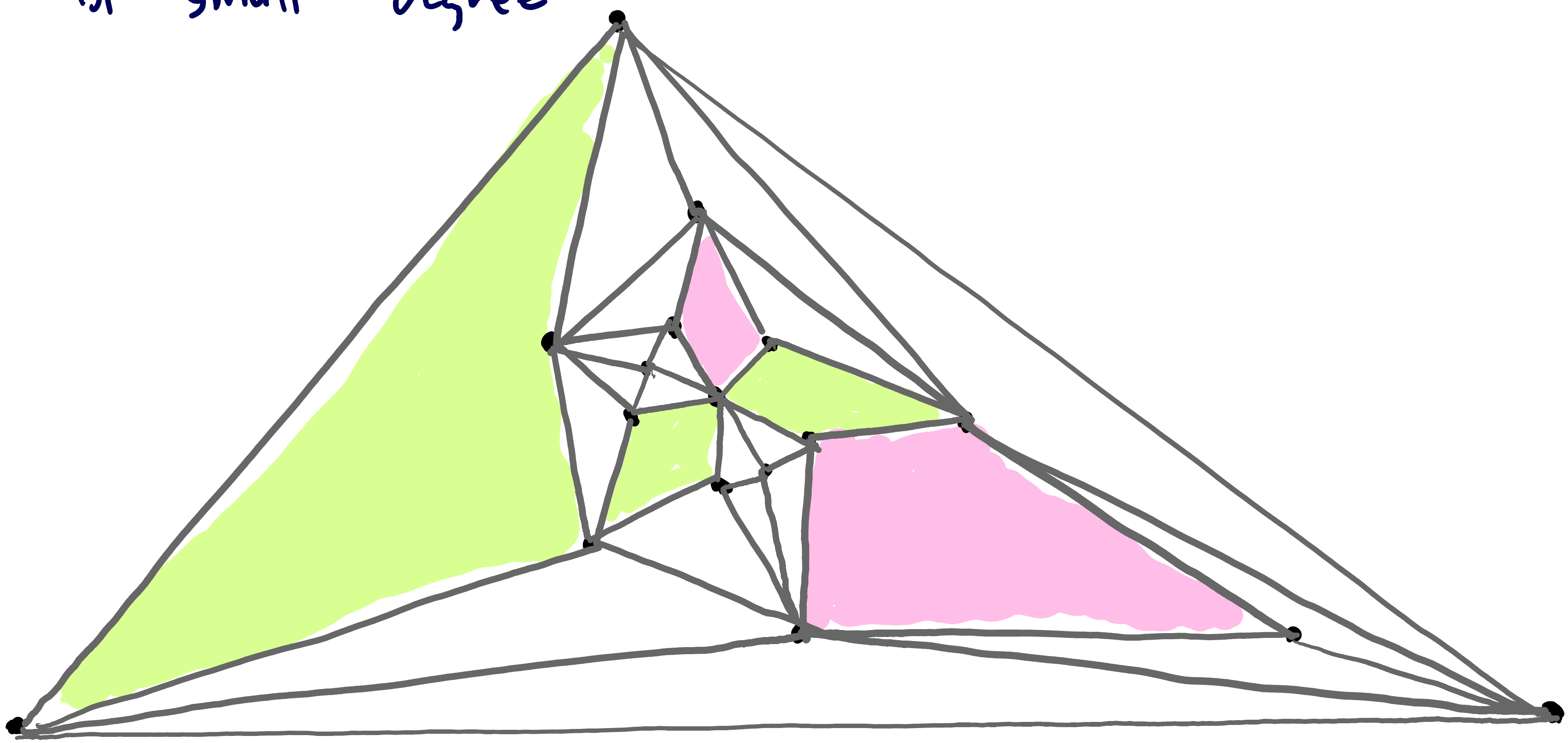
No! This is not a refinement, this is total destruction!



Remove an independent set of vertexes of small degree



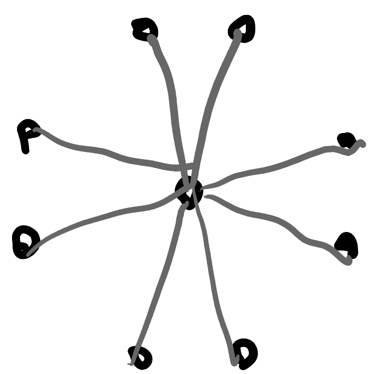
Remove an independent set of vertexes of small degree



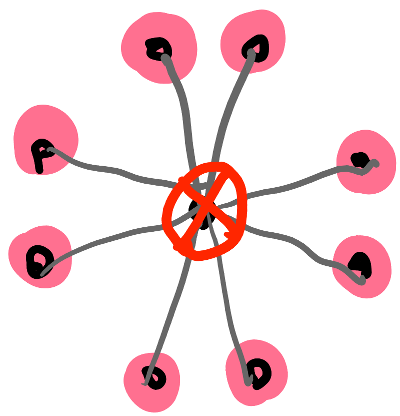
If half the vertexes have degree ≤ 8

how big of an independent set of degree ≤ 8 is there?

If half the vertexes have degree ≤ 8 how big of an independent set of degree ≤ 8 is there?



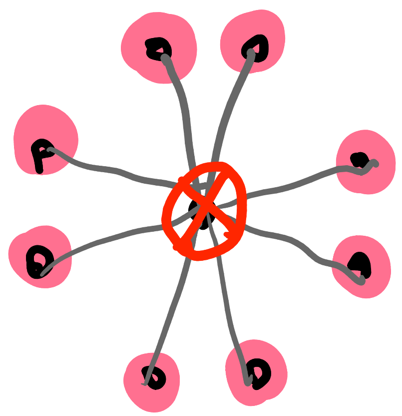
If half the vertexes have degree ≤ 8
how big of an independent set of degree ≤ 8 is there?



Remove unmarked vertex

Mark adjacent nodes as unremovable.

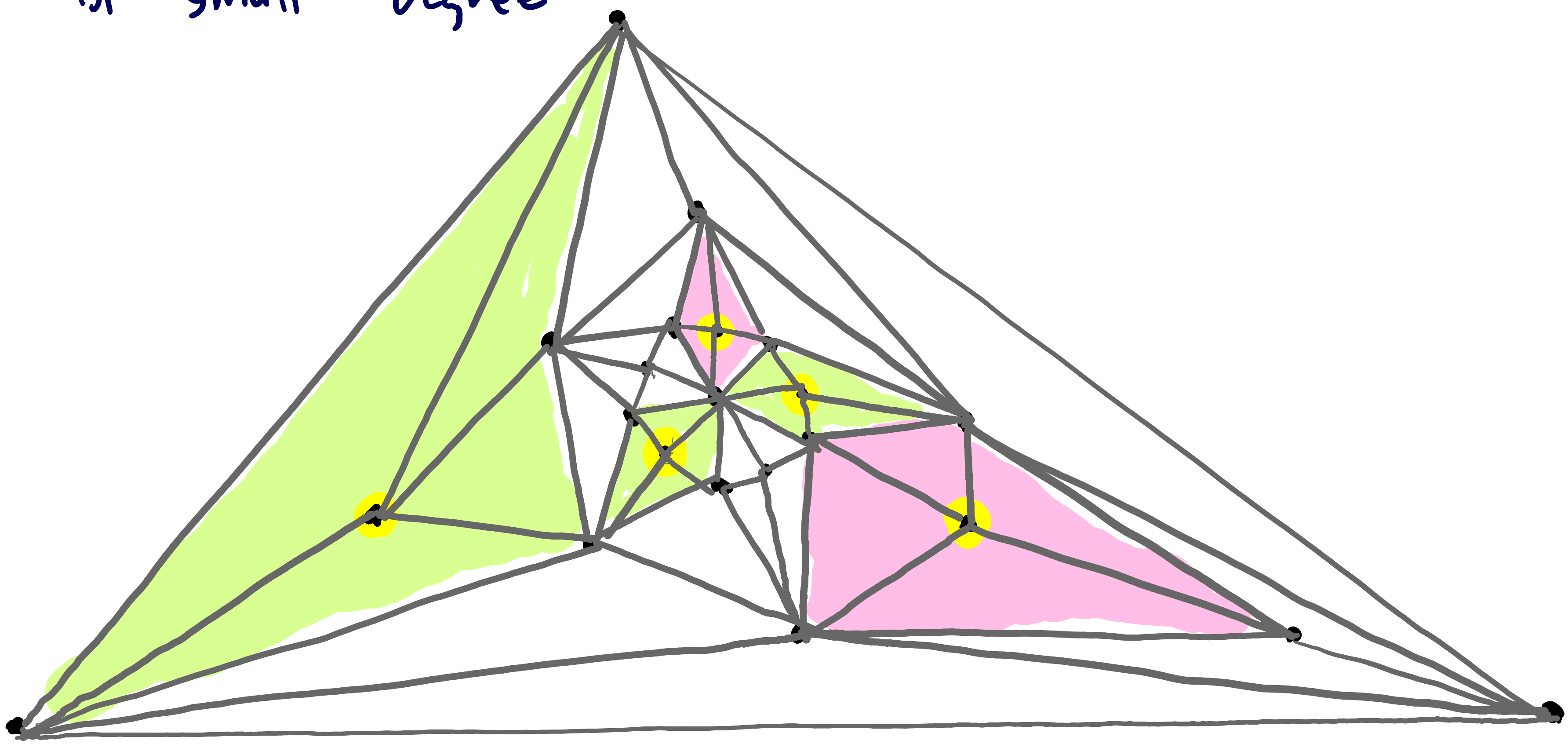
If half the vertexes have degree ≤ 8
how big of an independent set of degree ≤ 8 is there?



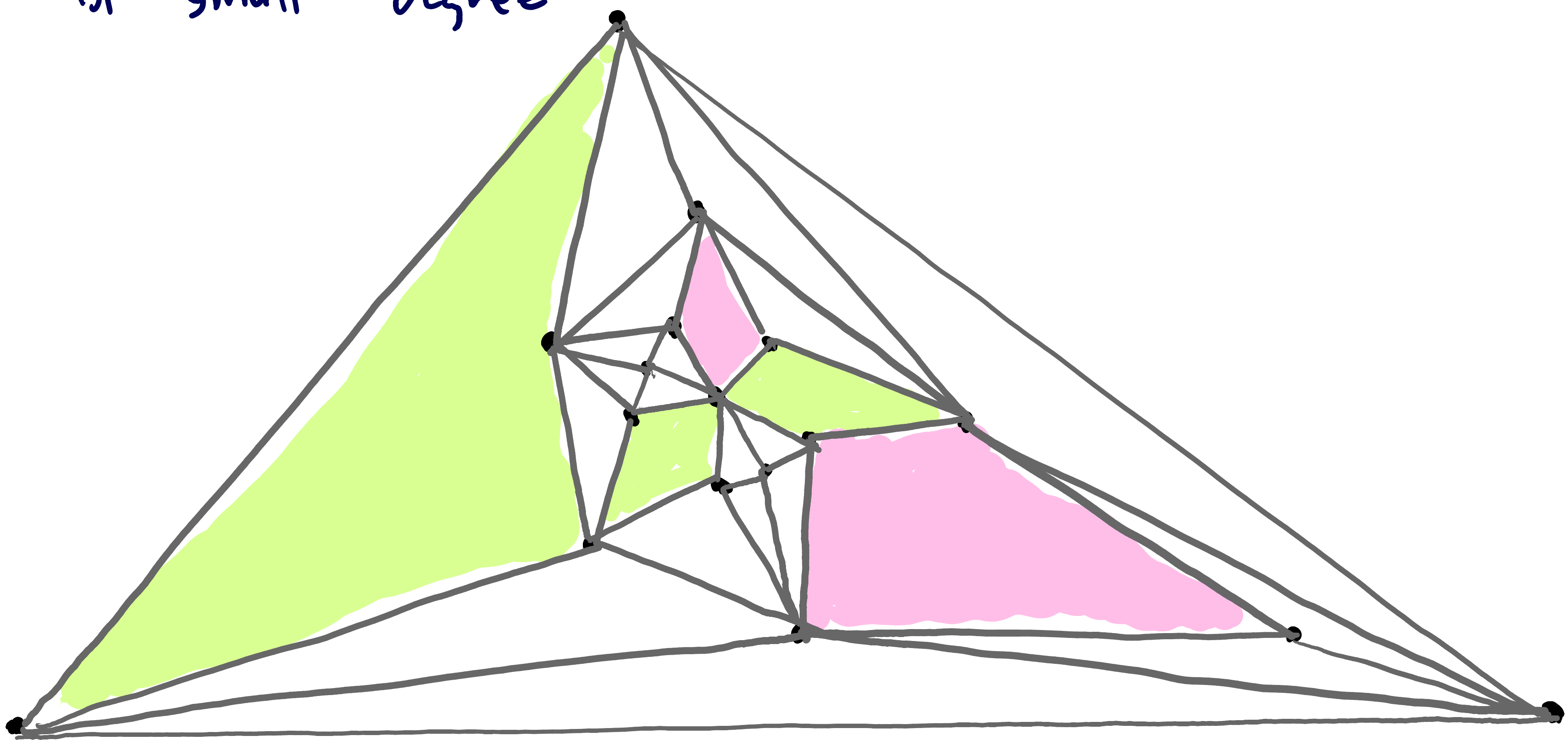
Remove unmarked vertex
Mark adjacent nodes as unremovable.

$$\frac{1}{9} \cdot \frac{1}{2} = \frac{1}{18}$$ of nodes will be removed

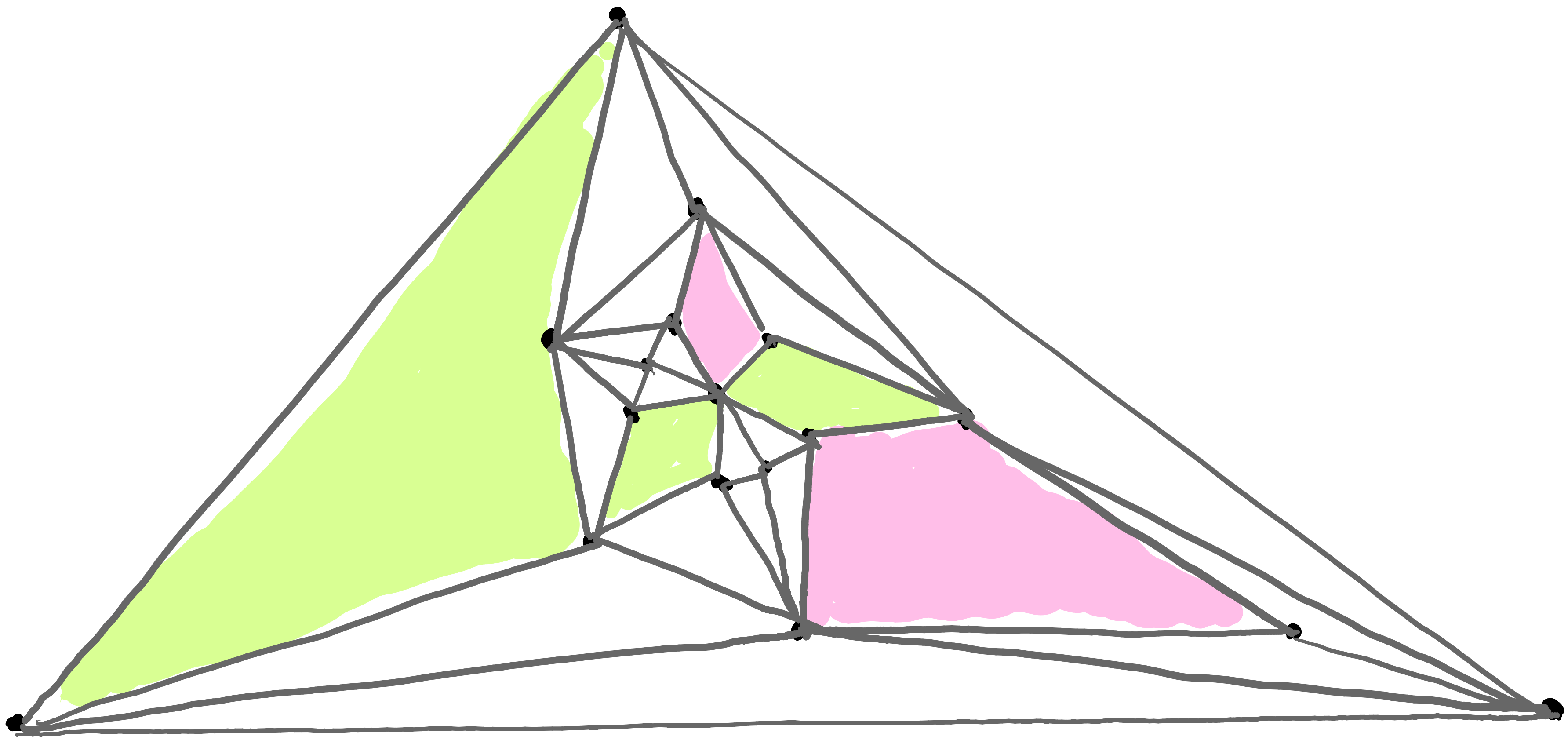
Remove an independent set of vertexes of small degree



Remove an independent set of vertexes of small degree

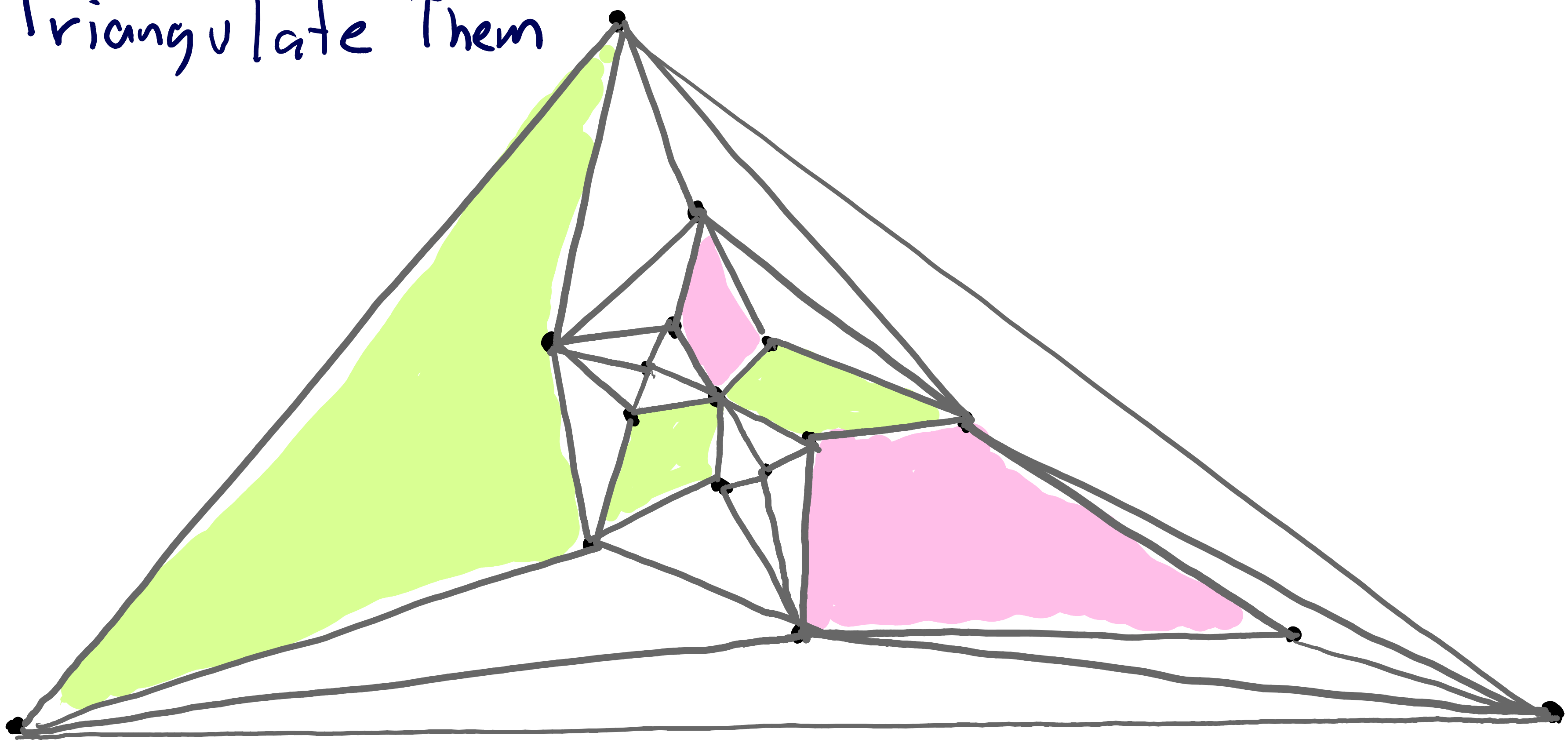


Resulting regions are not triangles



Resulting regions are not triangles

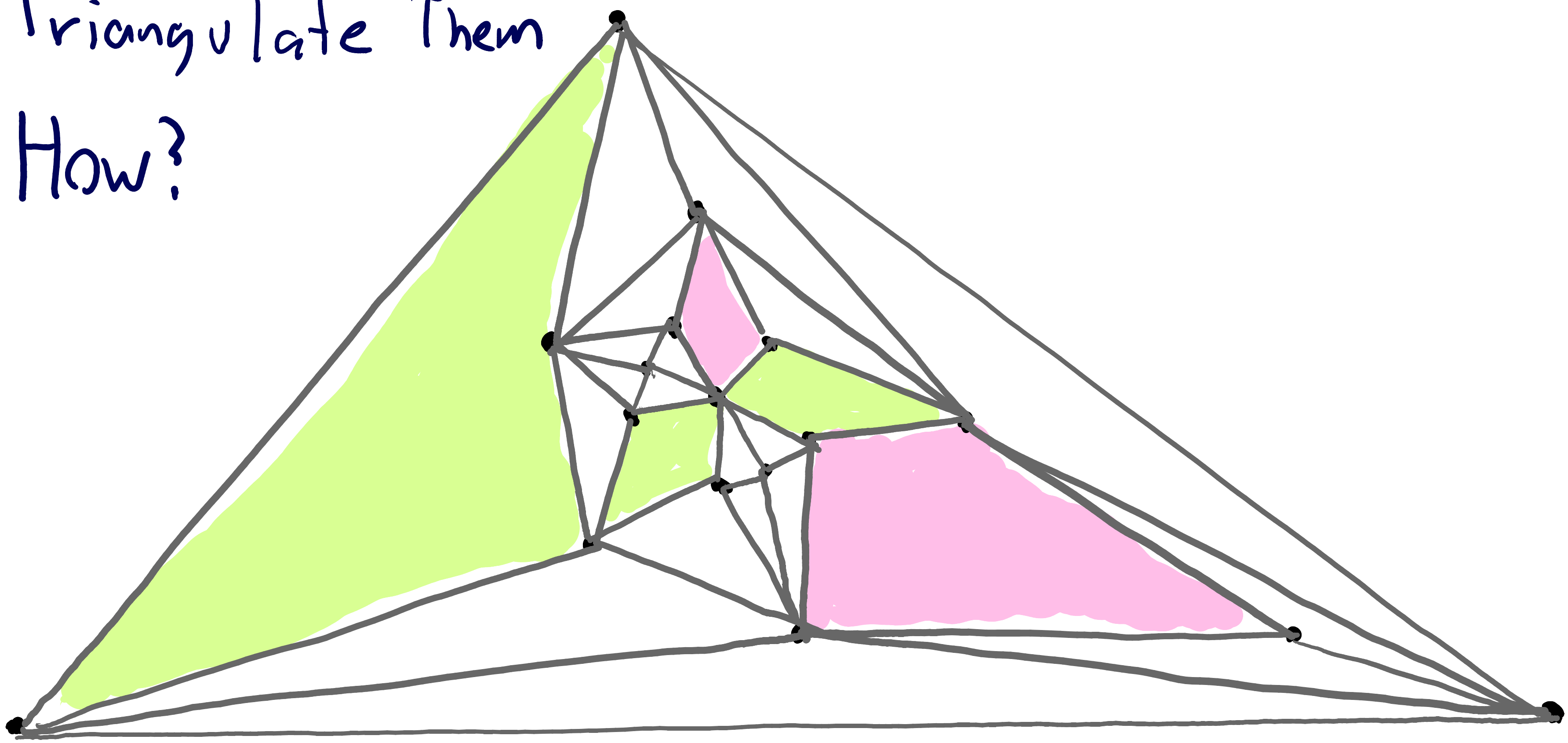
Triangulate Them



Resulting regions are not triangles

Triangulate Them

How?

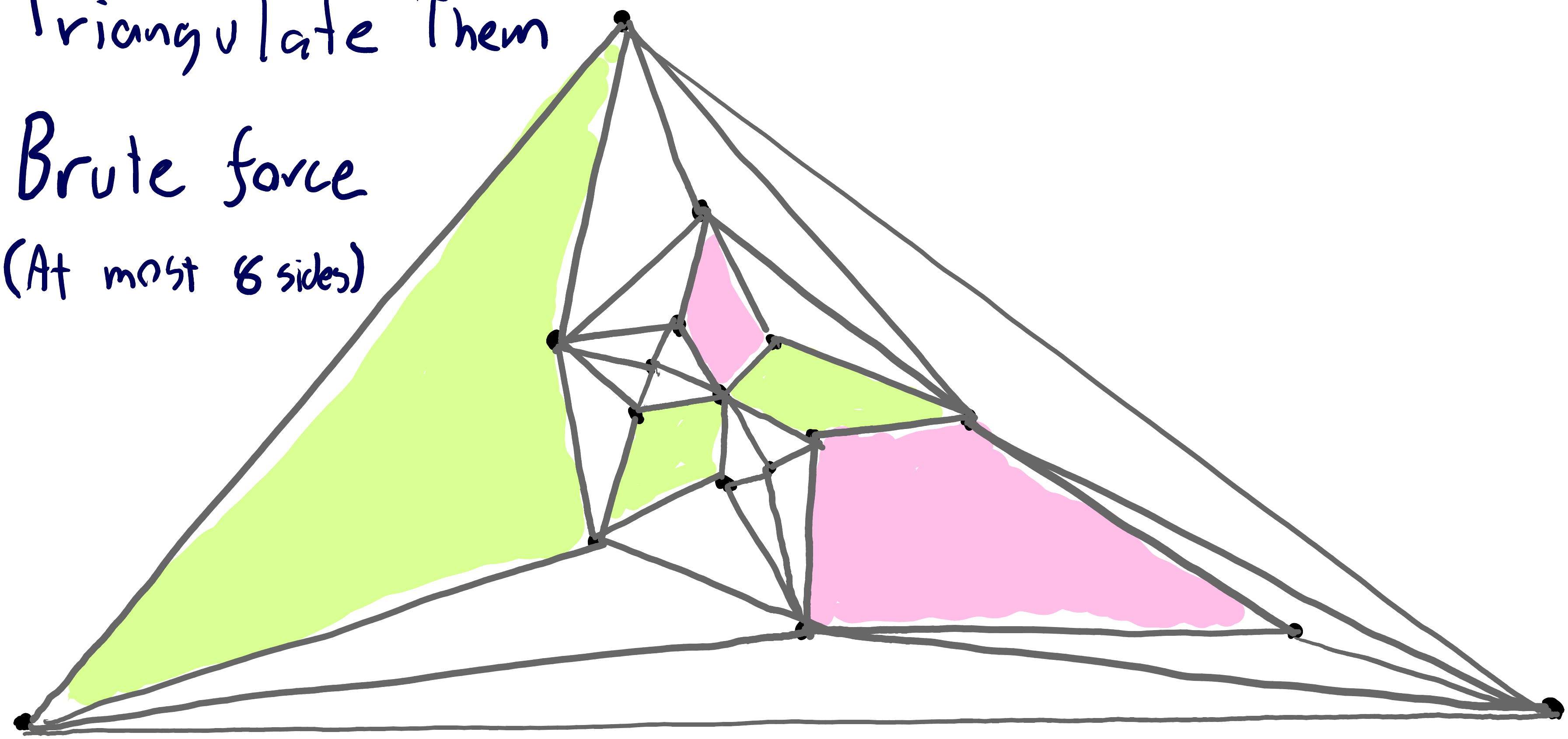


Resulting regions are not triangles

Triangulate Them

Brute force

(At most 8 sides)

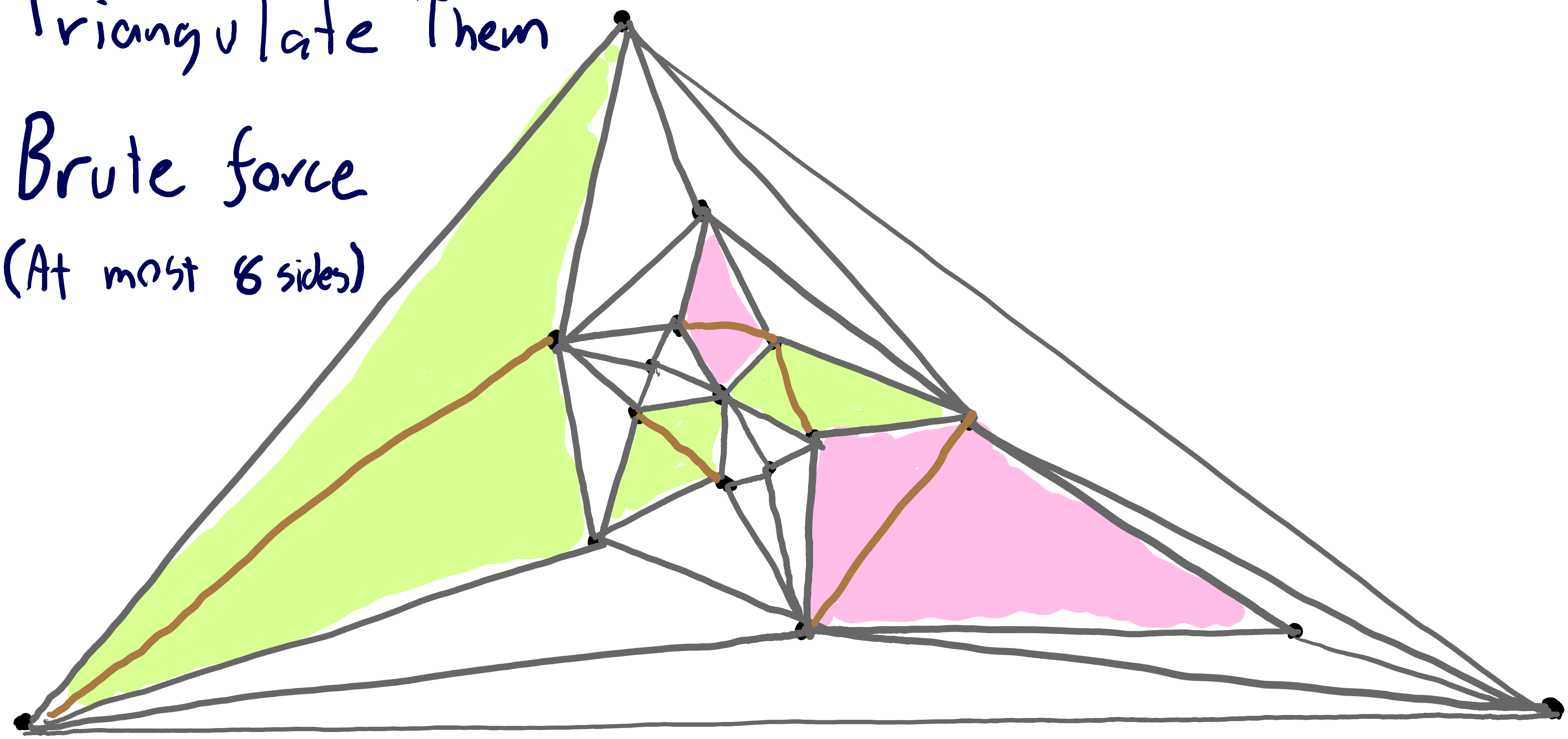


Resulting regions are not triangles

Triangulate Them

Brute force

(At most 8 sides)



Properties of this construction

Properties of this construction

(1) New triangulation has $\leq \frac{17}{18} \cdot$ vertexes of old triangulation

Properties of this construction

(1) New triangulation has $\leq \frac{17}{18} \cdot$ vertexes of old triangulation

(2) New triangulation can be computed in linear time

Properties of this construction

(1) New triangulation has $\leq \frac{17}{18} \cdot$ vertexes of old triangulation

(2) New triangulation can be computed in linear time

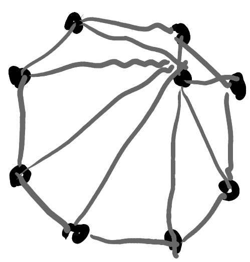
(3) Every triange in the new triangulation overlaps ≤ 8 trianges in the old triangulation

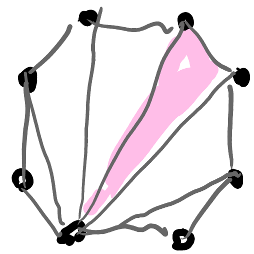
Properties of this construction

(1) New triangulation has $\leq \frac{17}{18} \cdot$ vertexes of old triangulation

(2) New triangulation can be computed in linear time

(3) Every triange in the new triangulation overlaps ≤ 8 trianges in the old triangulation



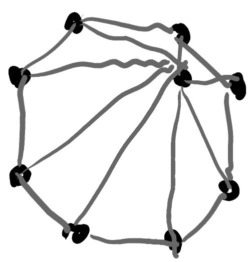


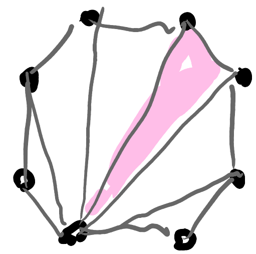
Properties of this construction

(1) New triangulation has $\leq \frac{17}{18} \cdot$ vertexes of old triangulation

(2) New triangulation can be computed in linear time

(3) Every triange in the new triangulation overlaps ≤ 8 trianges in the old triangulation





(4) Put a pointer from every triange in the OT to every triangle in the NT it overlaps.

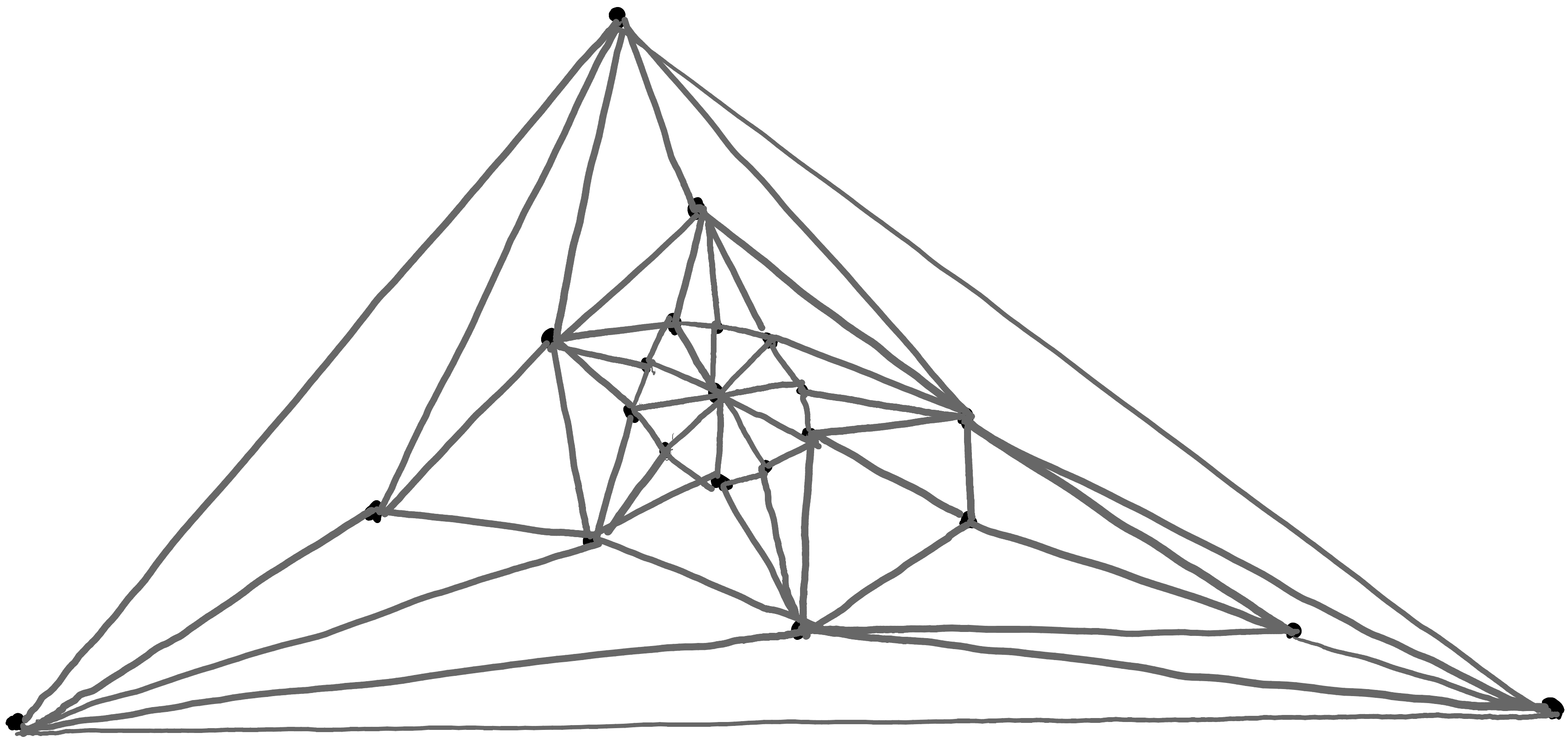
Then given a point and the triange in the OT which it is in the triangle in NT that it is in can be determined in $O(1)$.

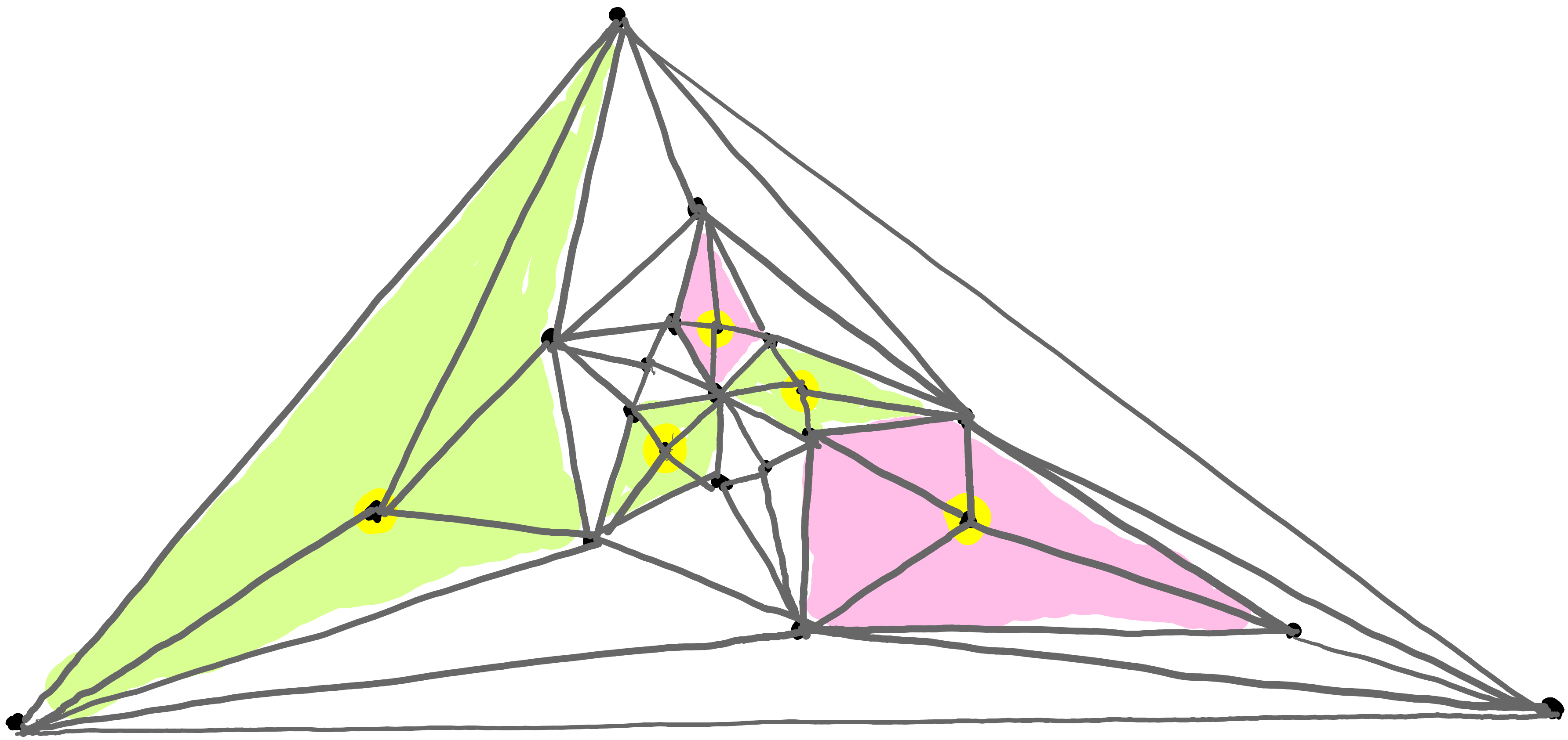
This gives you everything you need

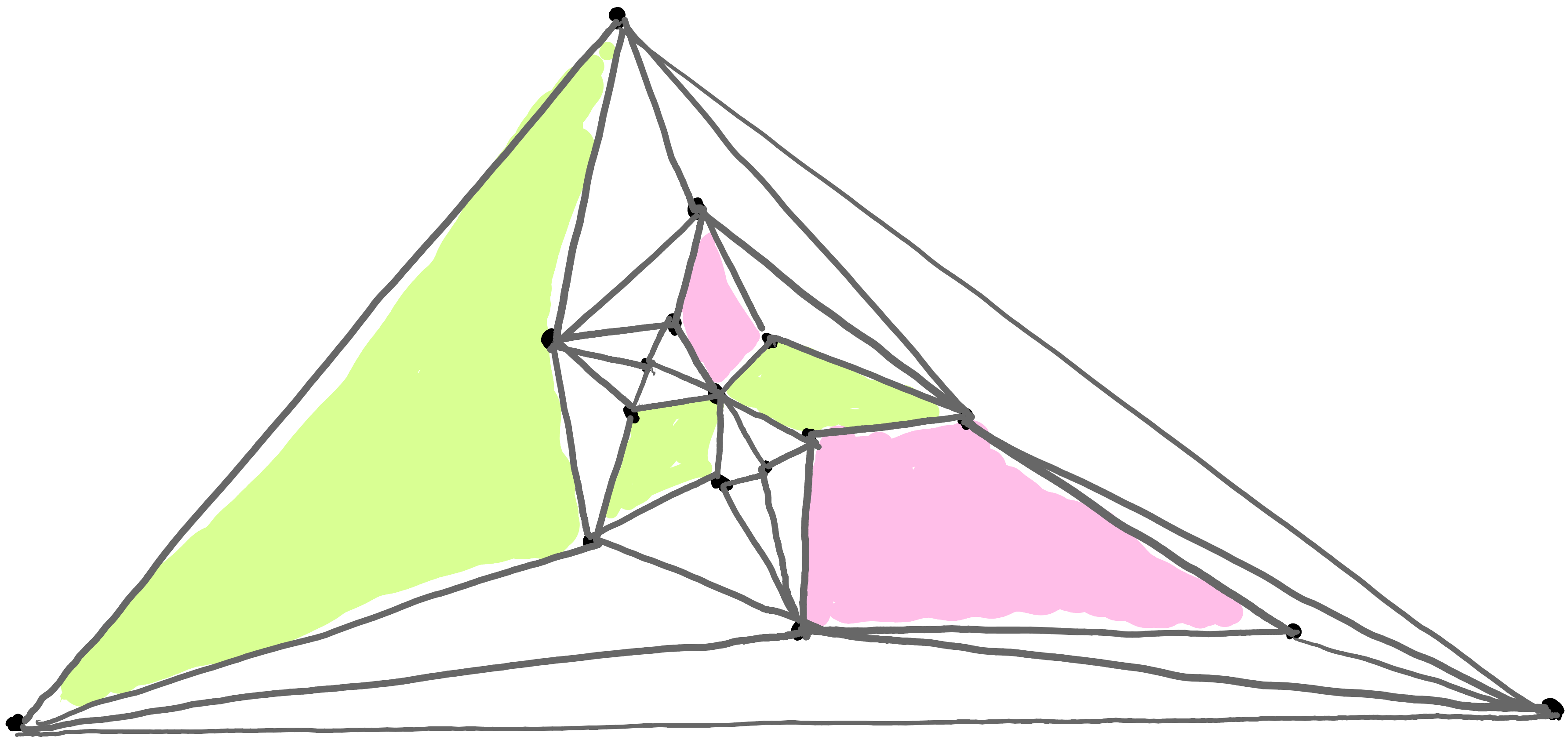
Just repeat the

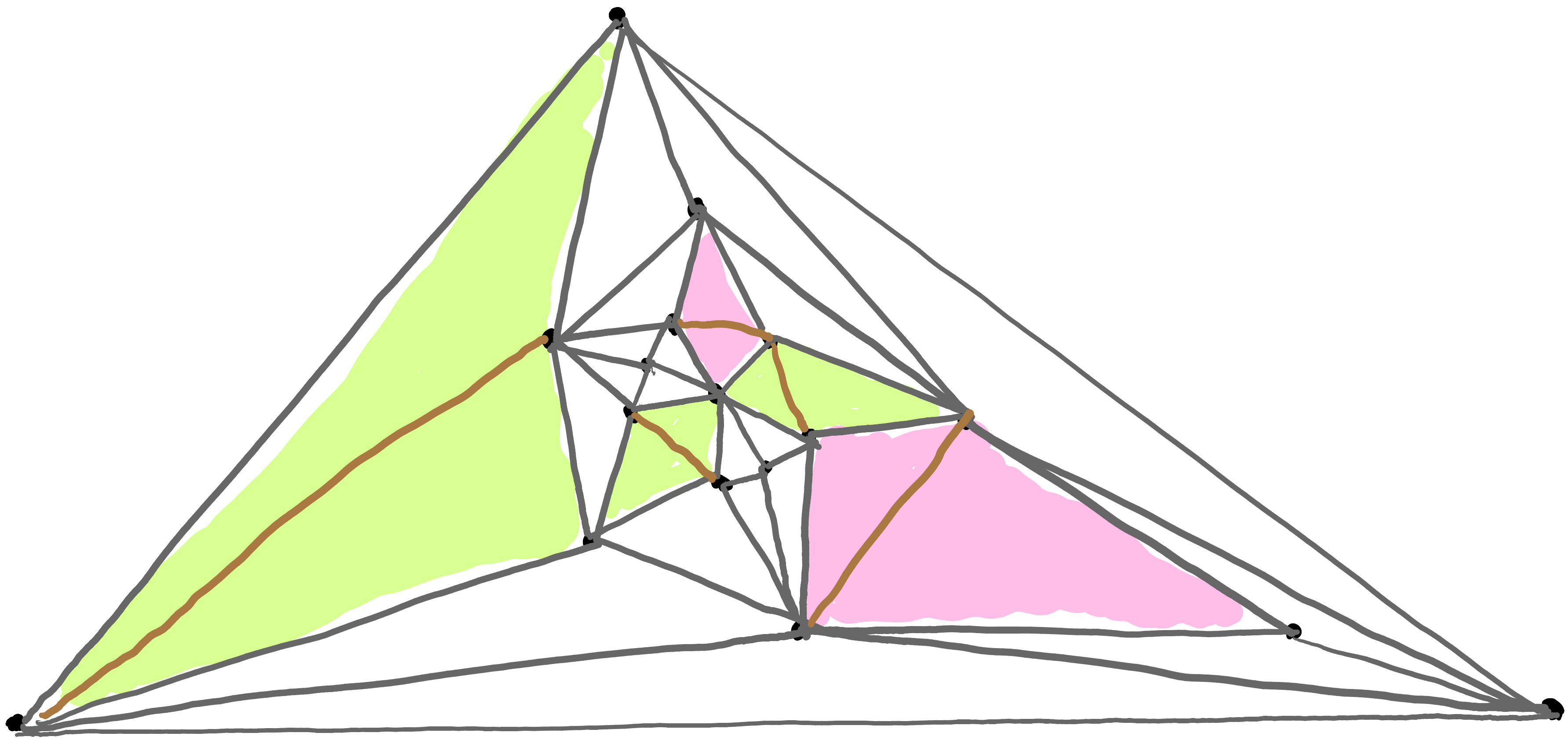
refinement

Level 0

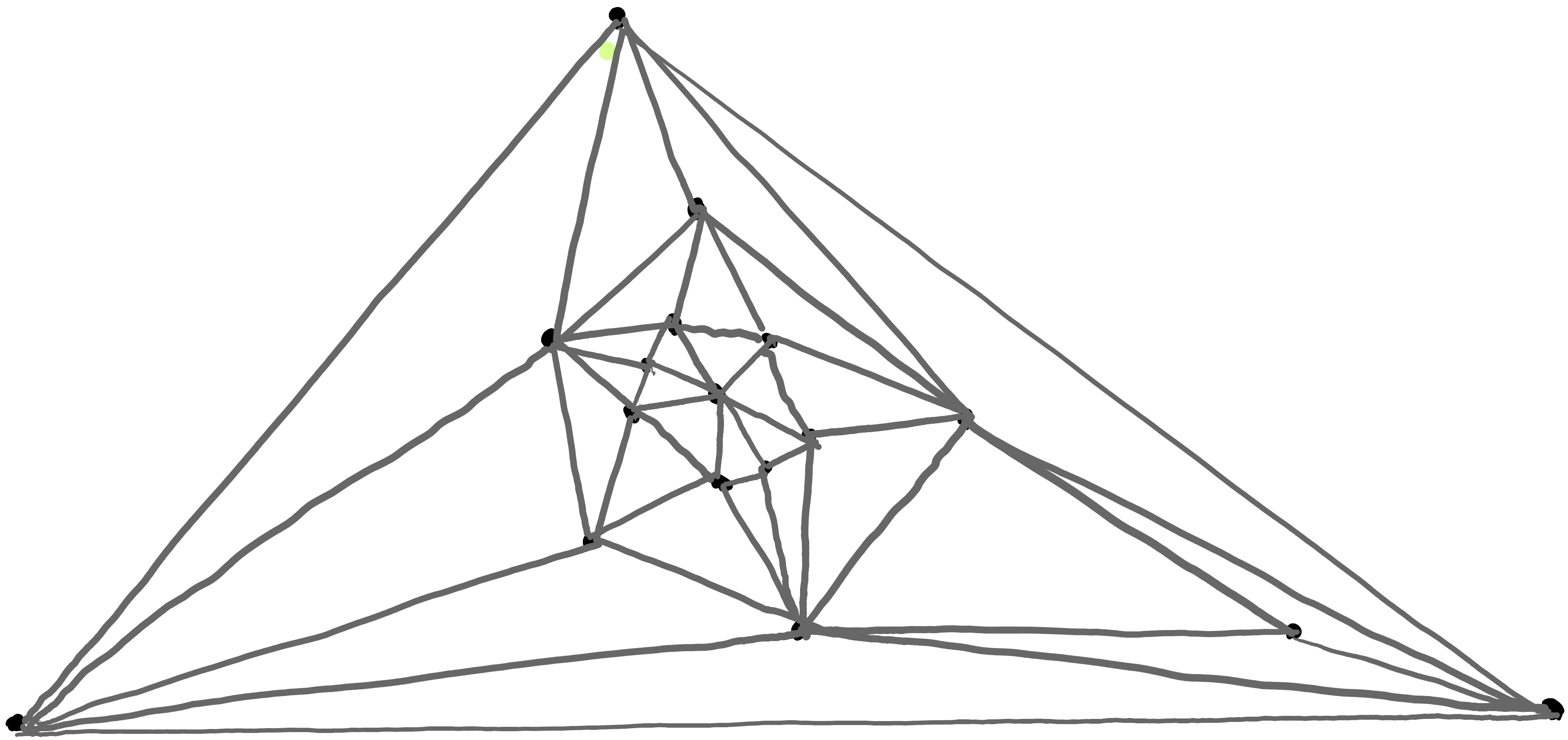


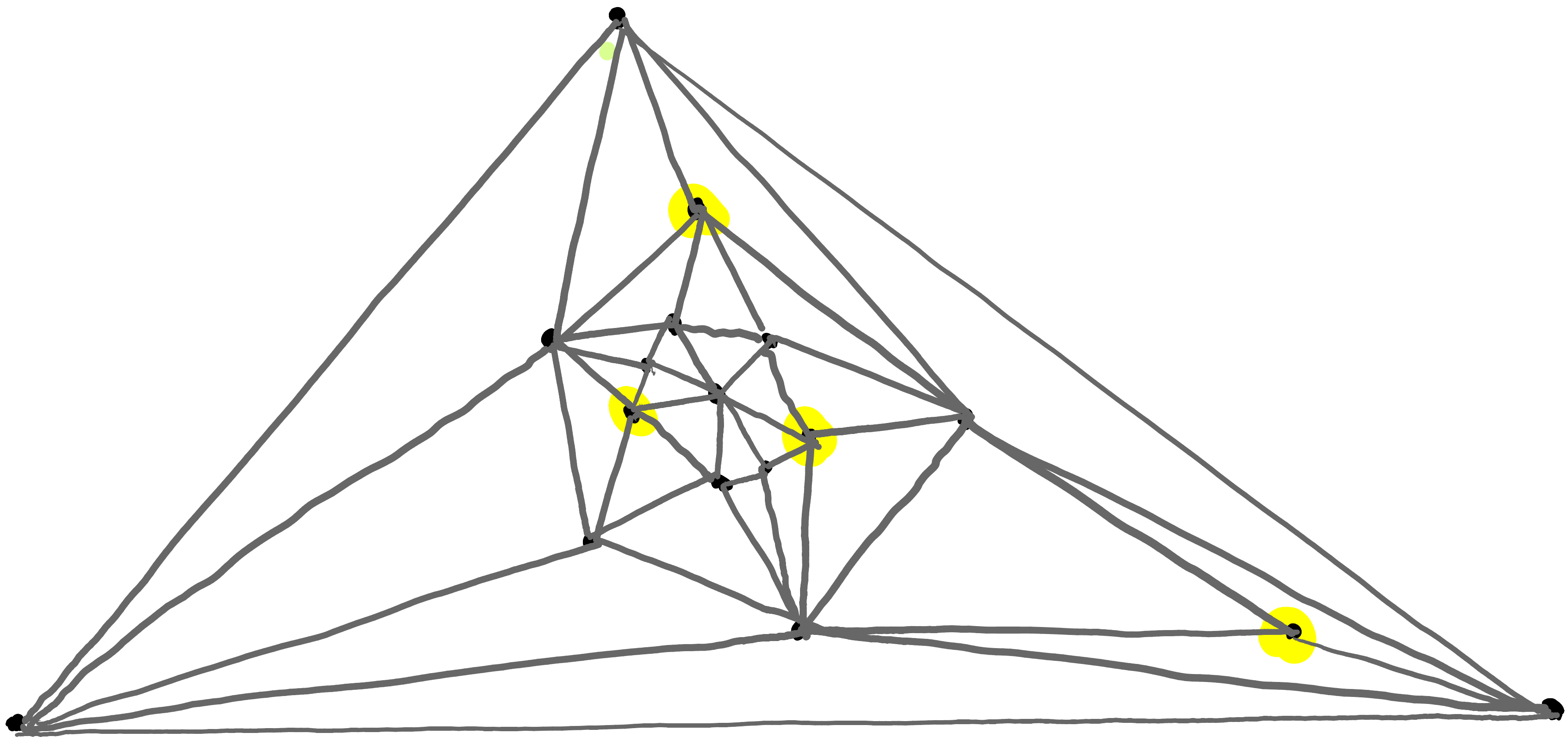


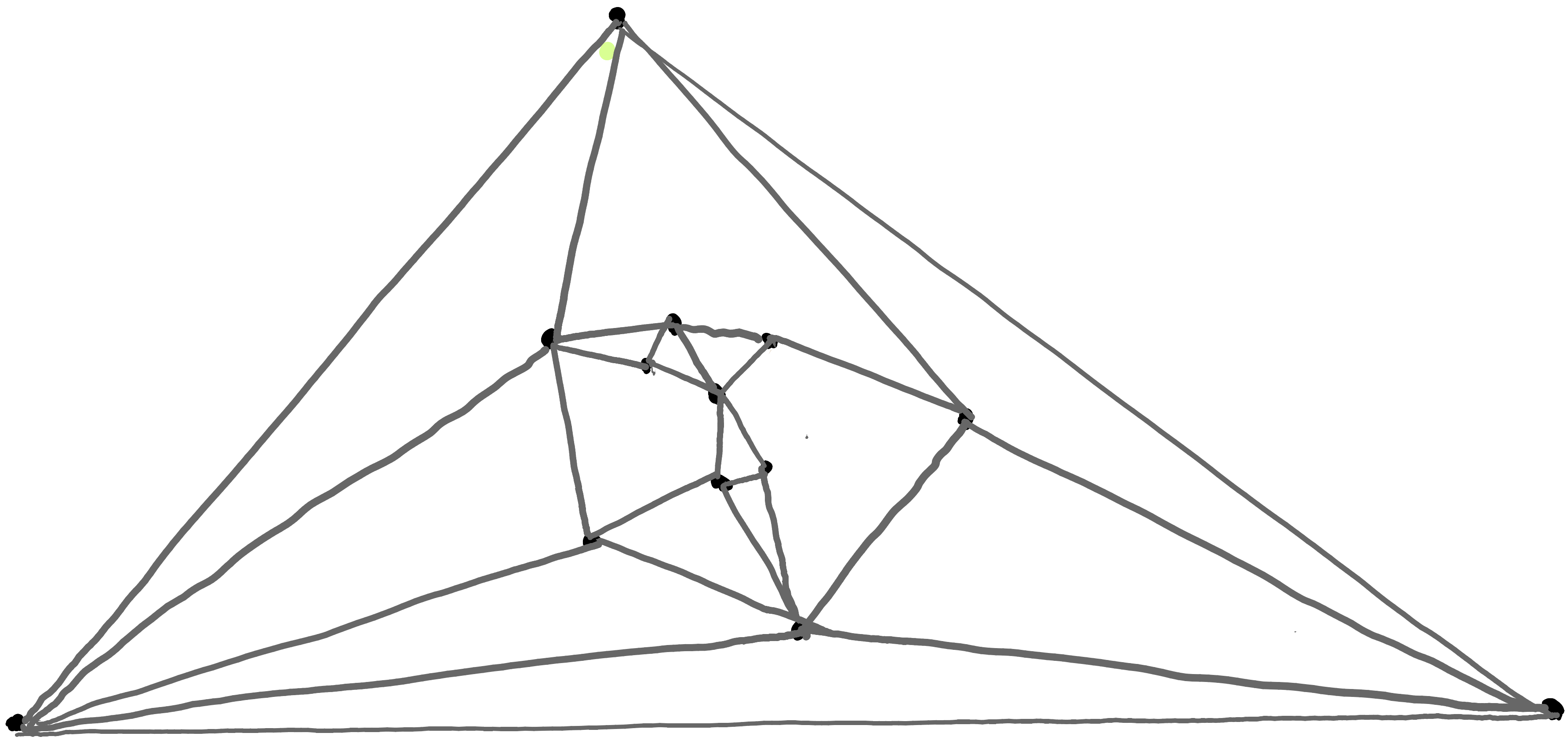


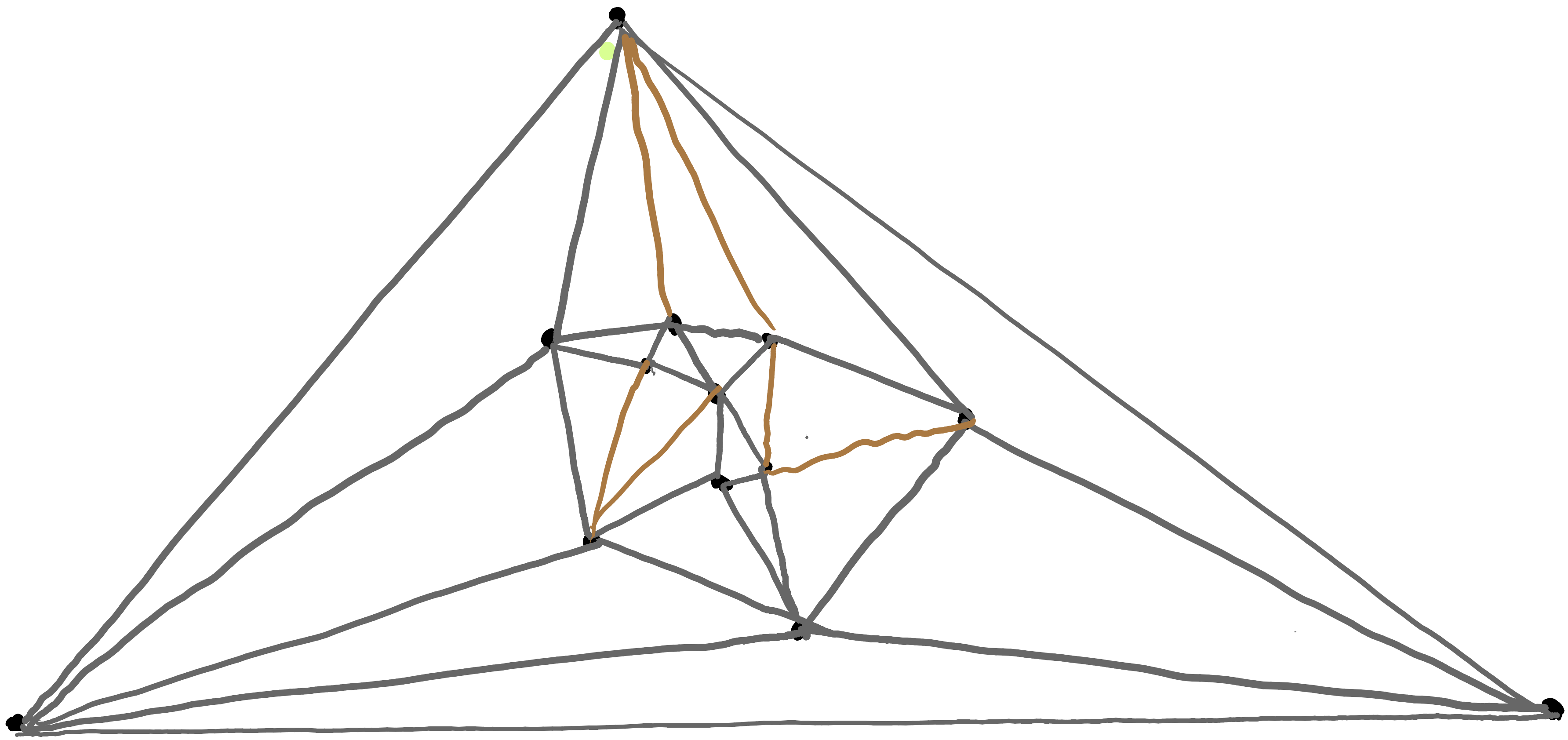


Level 1

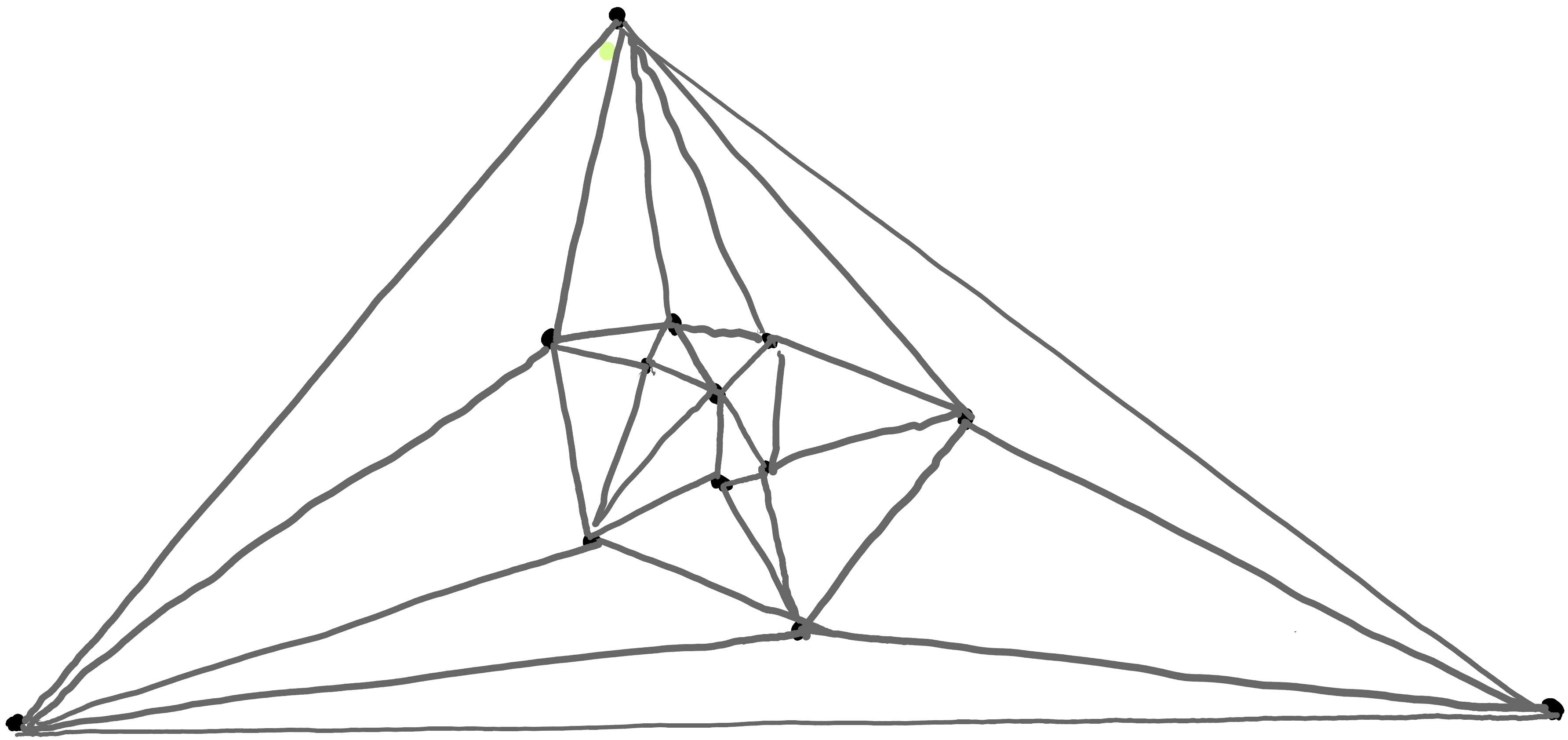


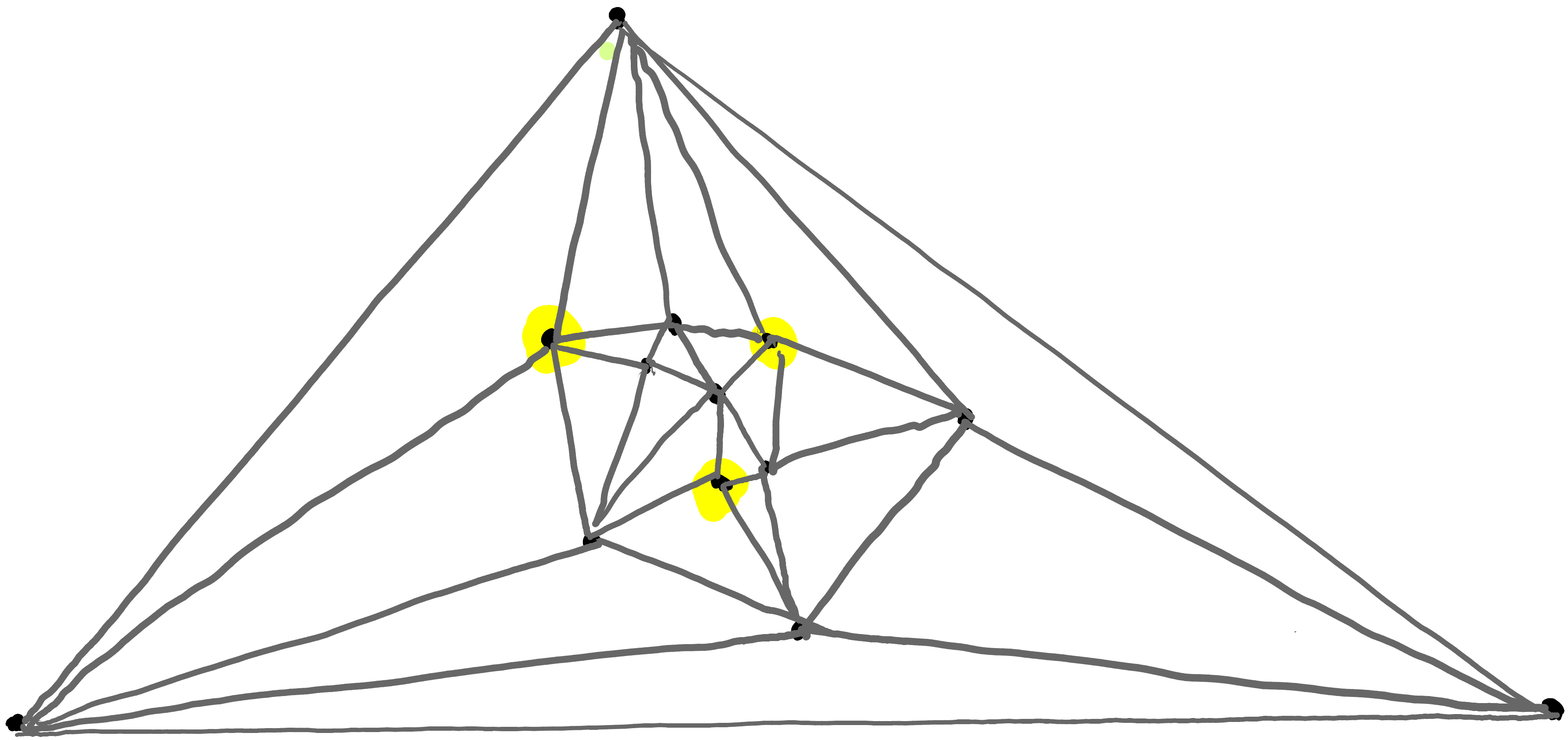


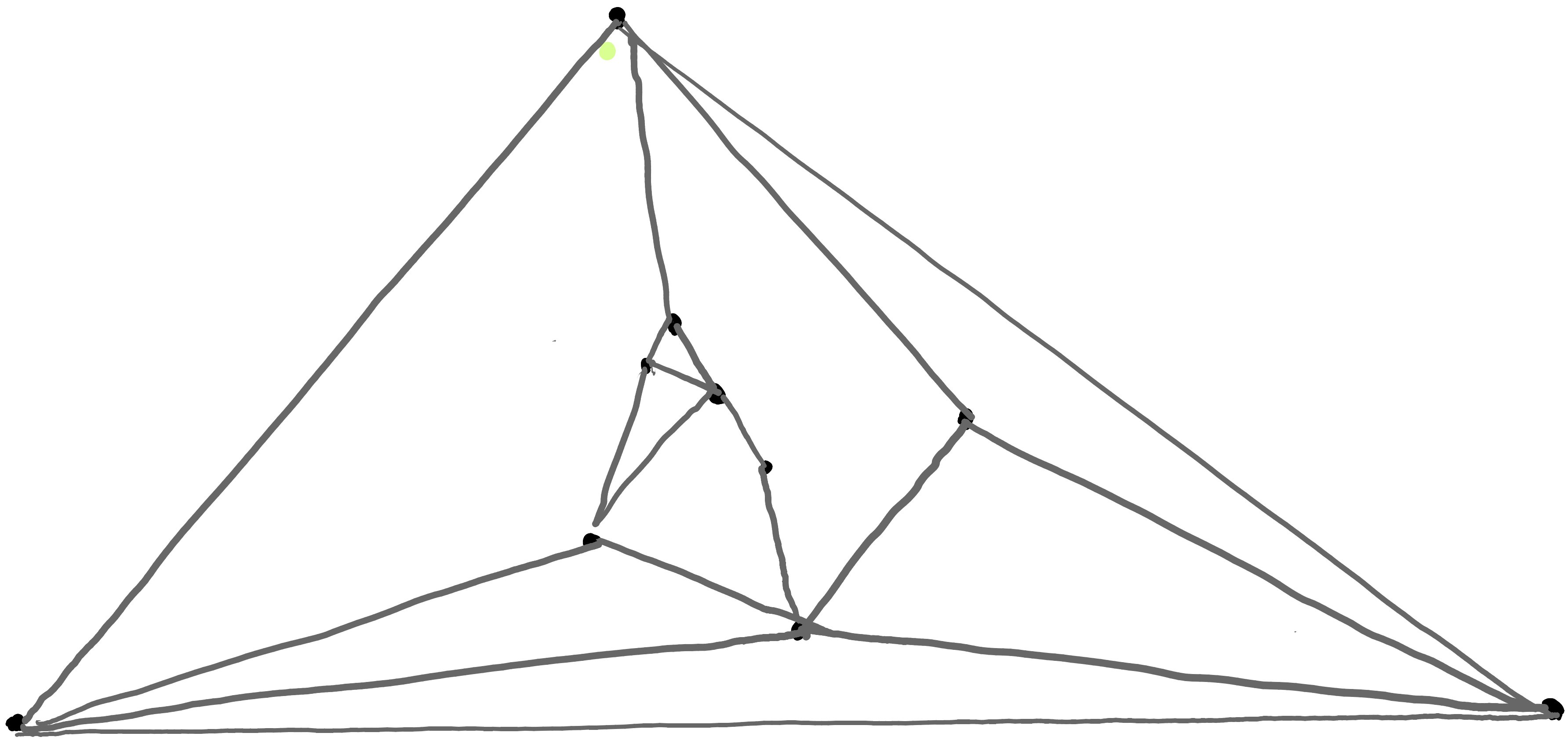




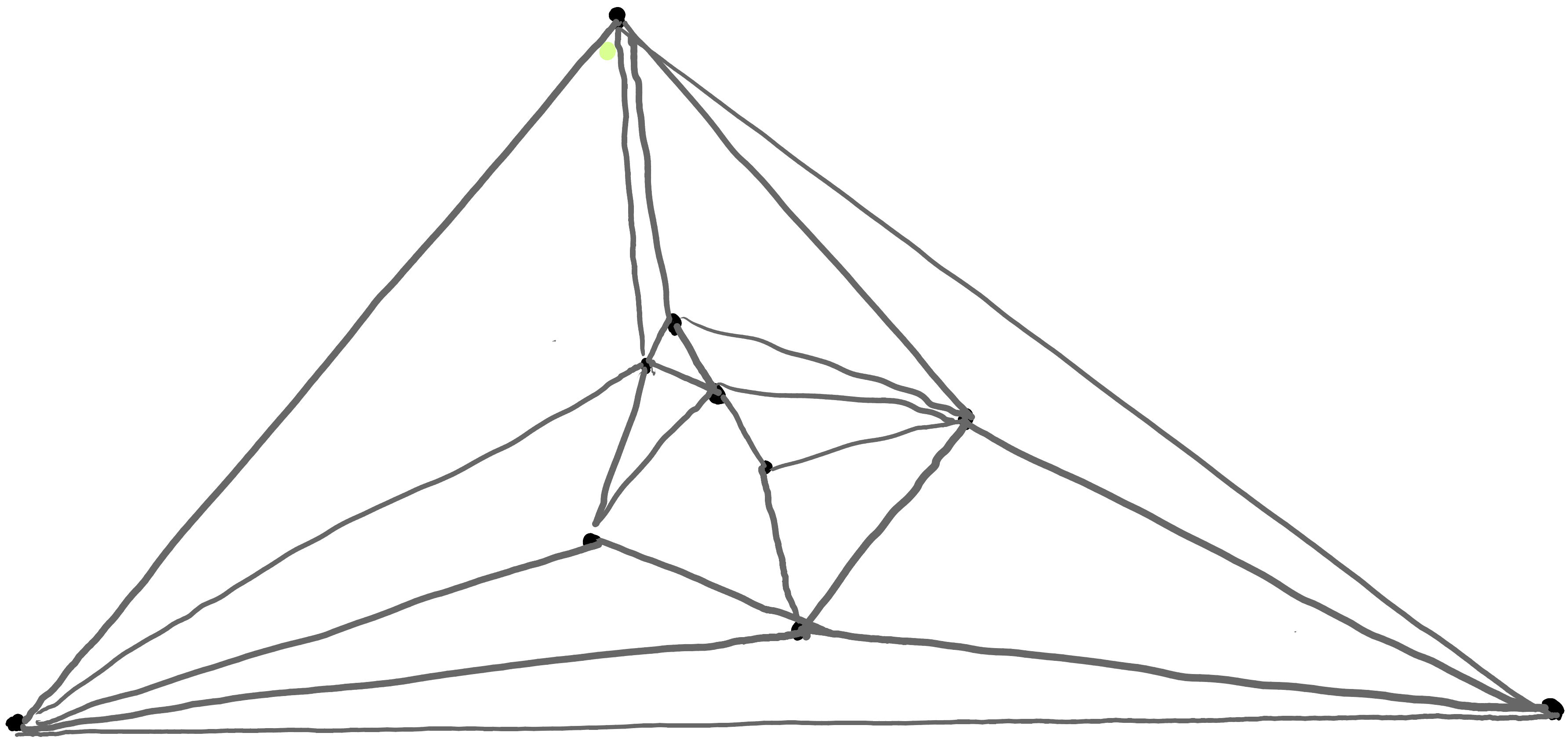
Level 2

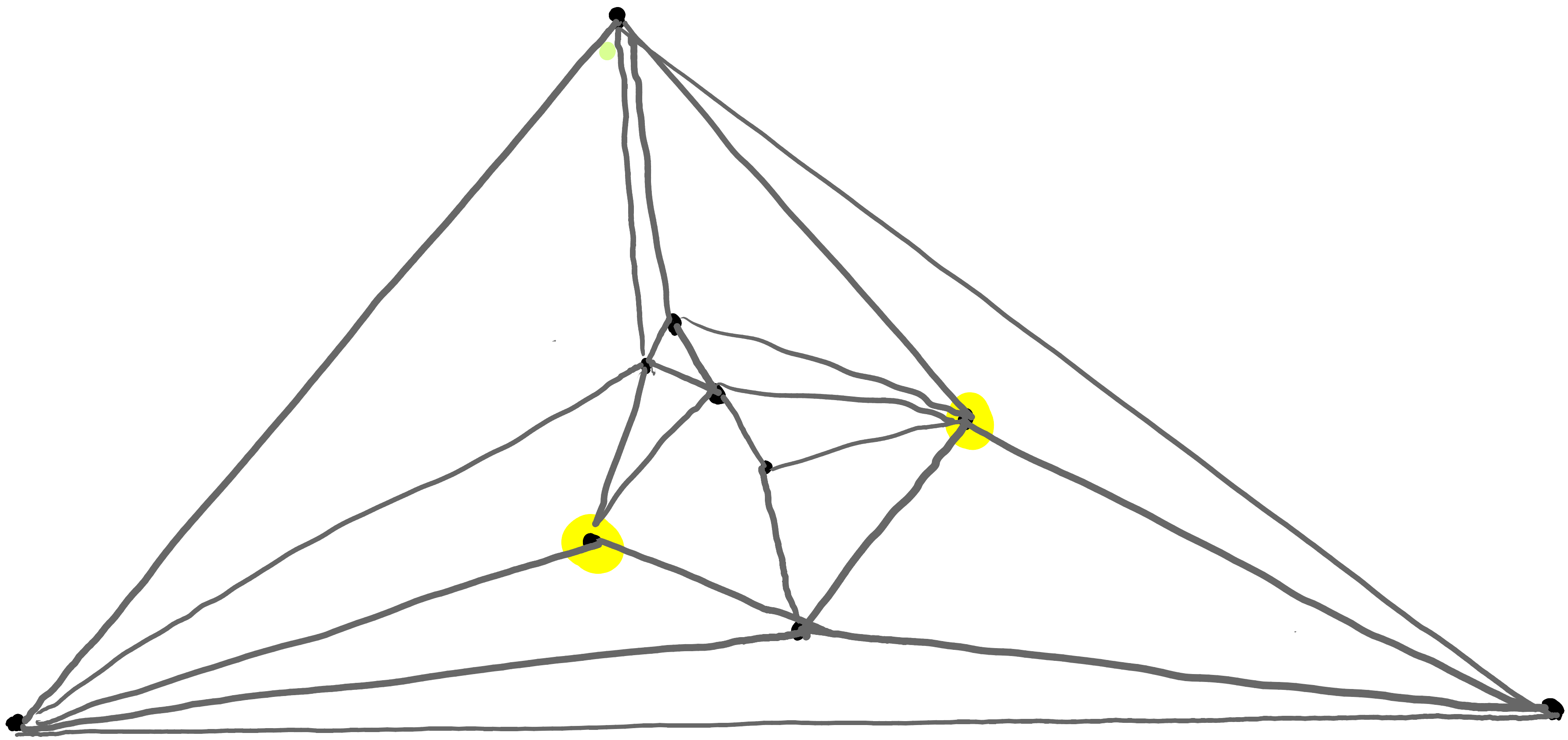


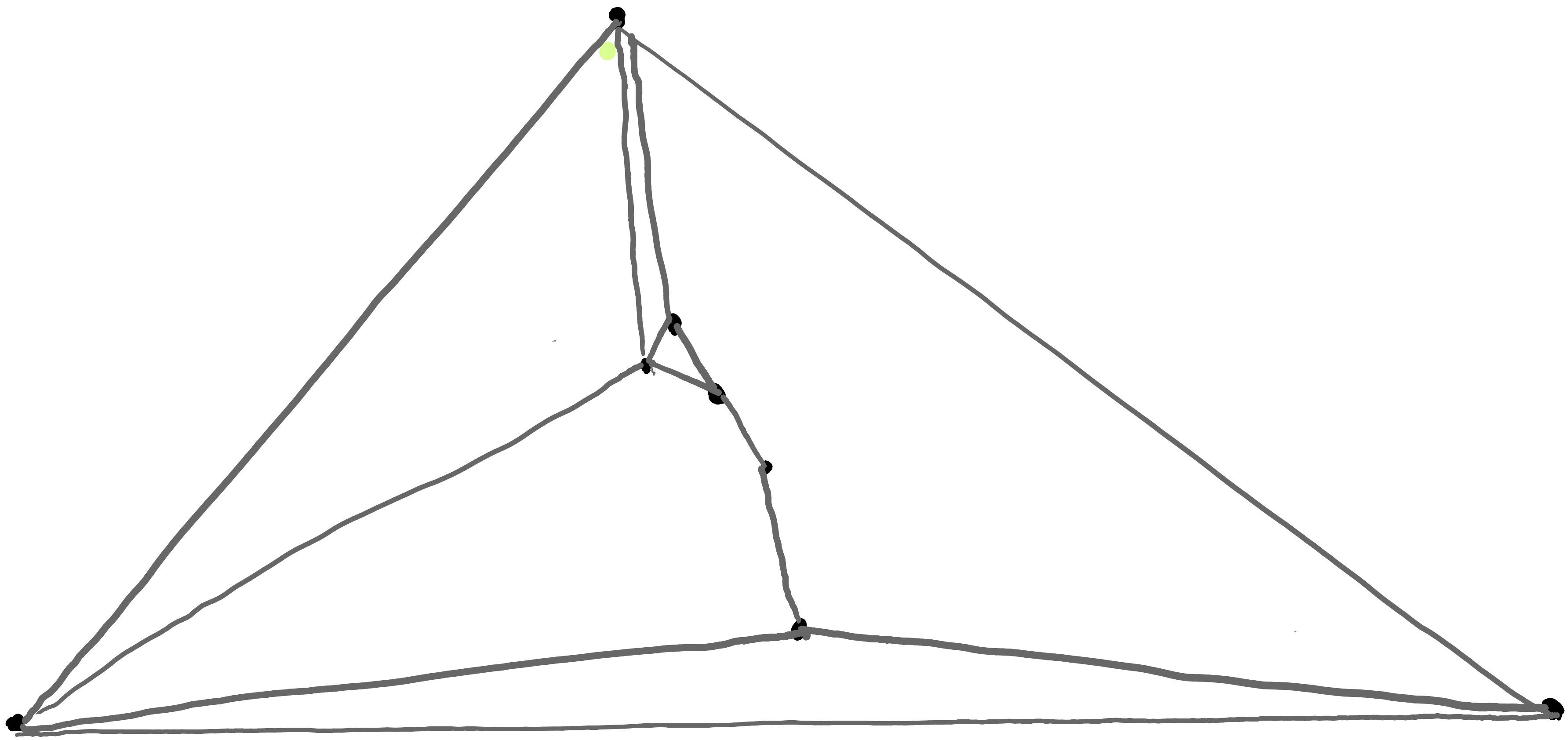




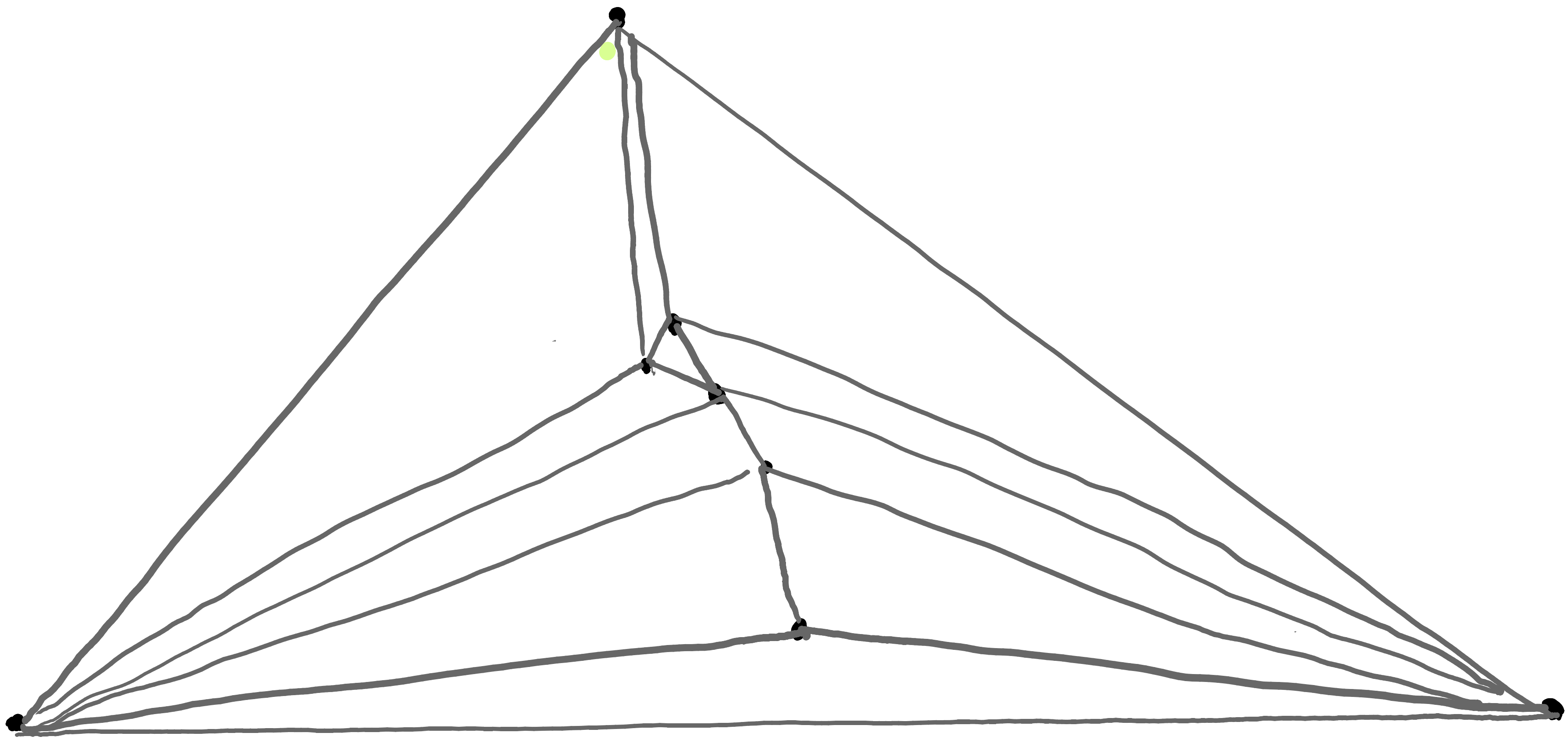
Level 3

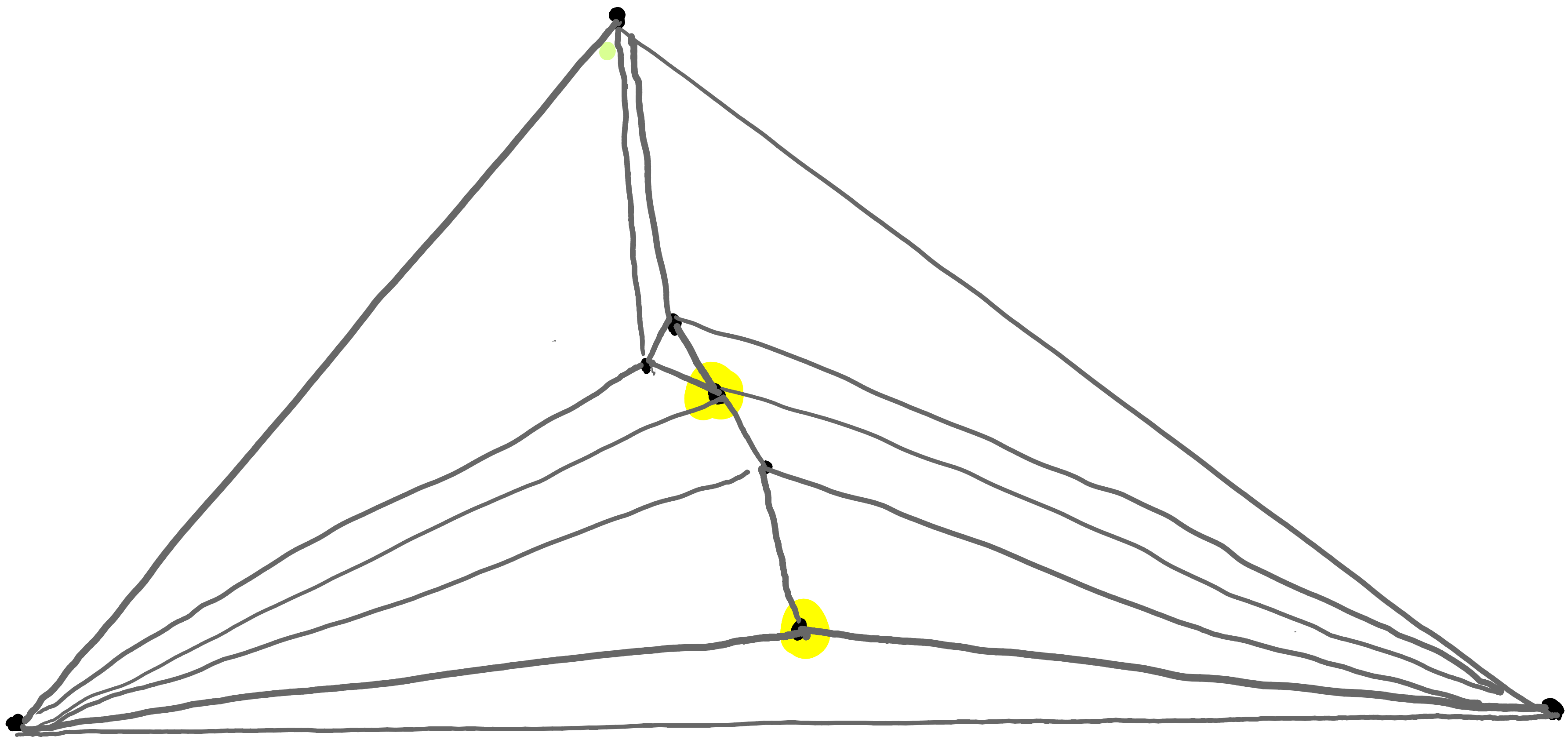


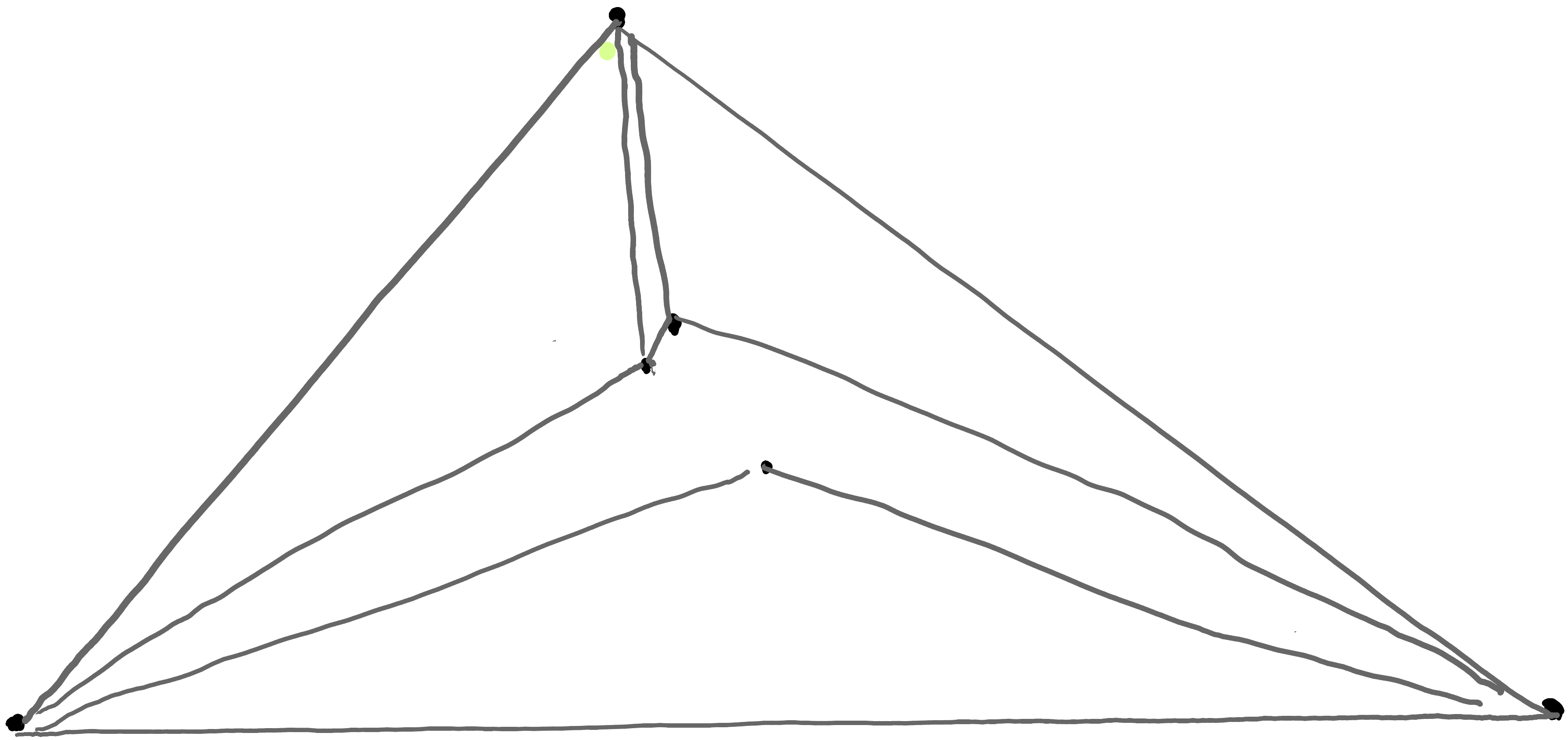




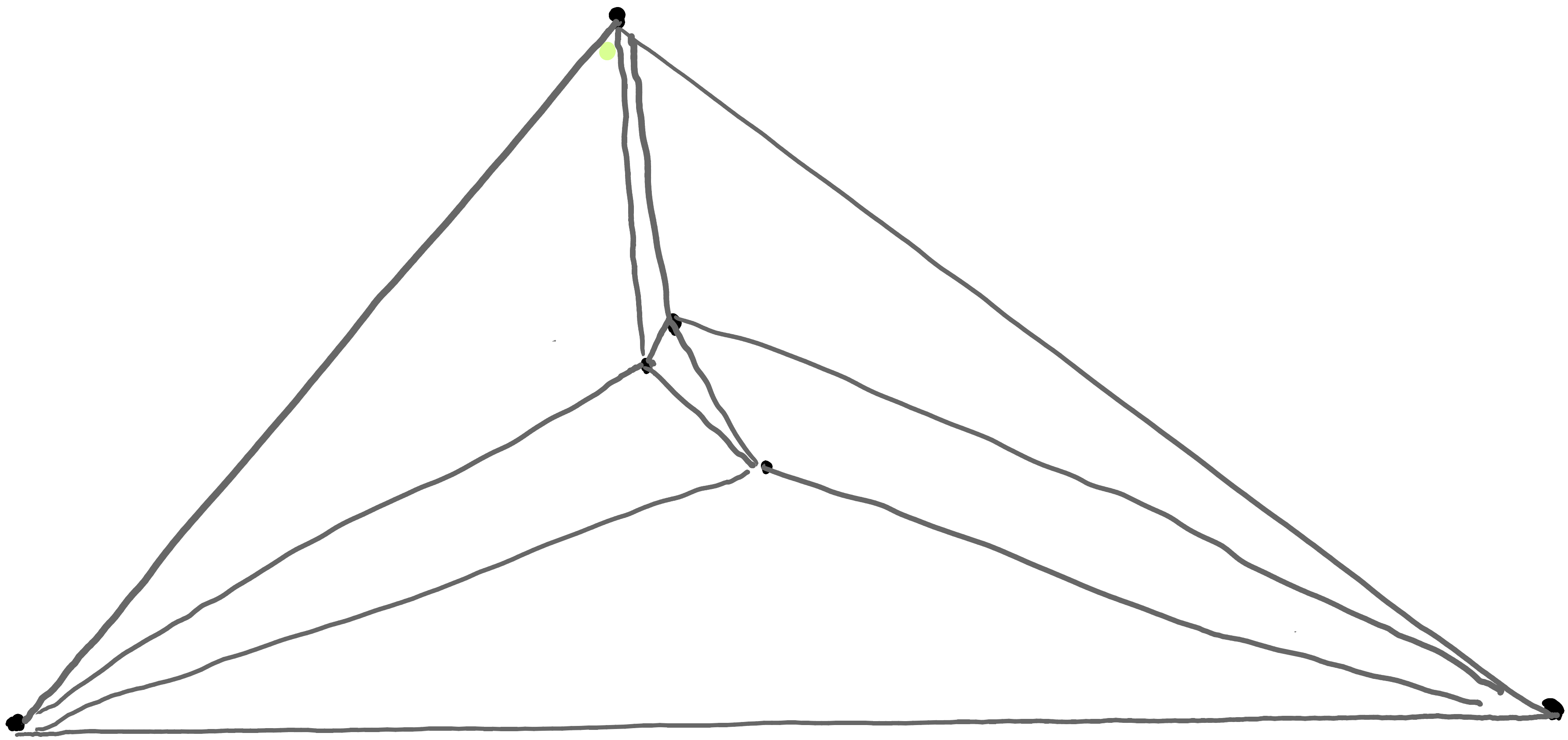
Level 4

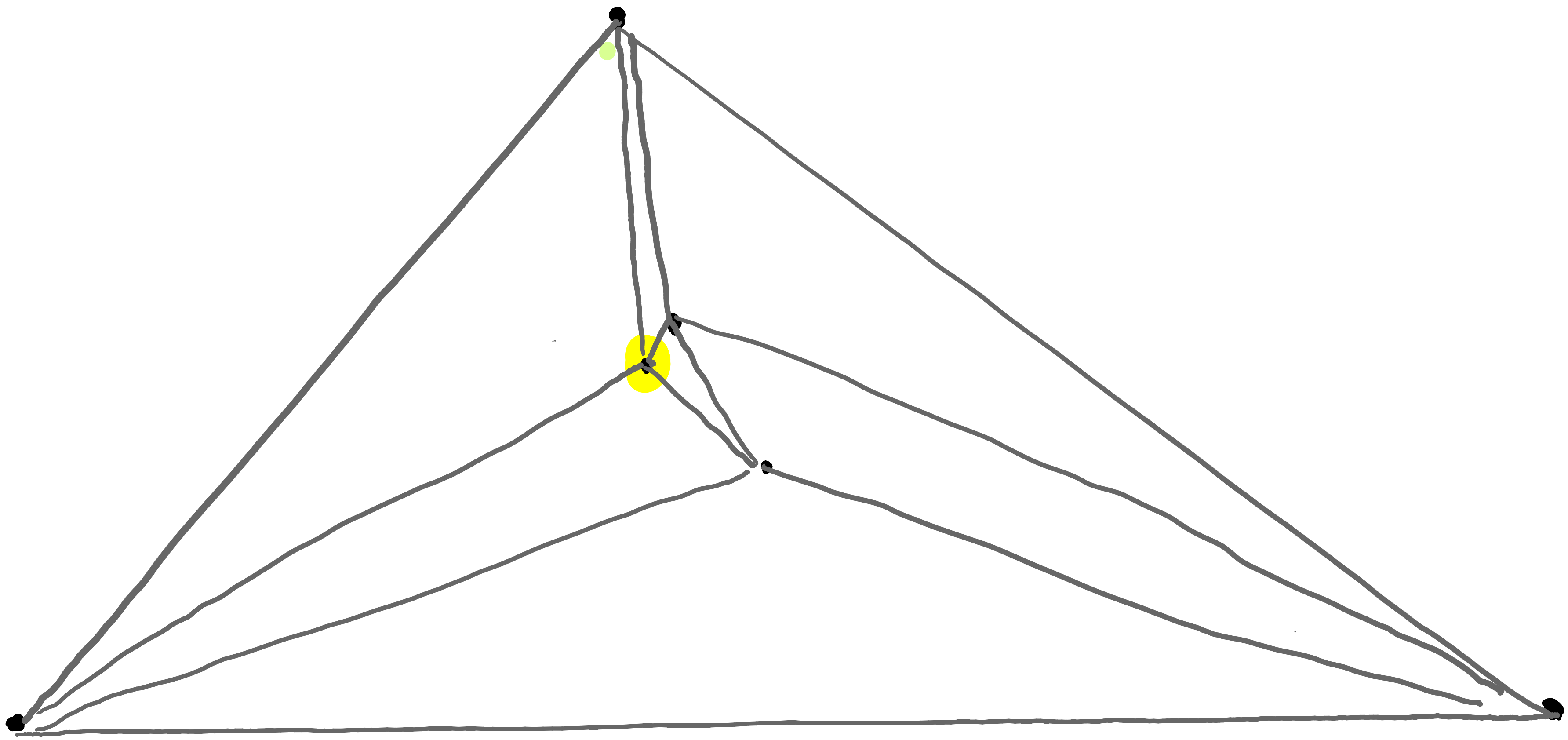


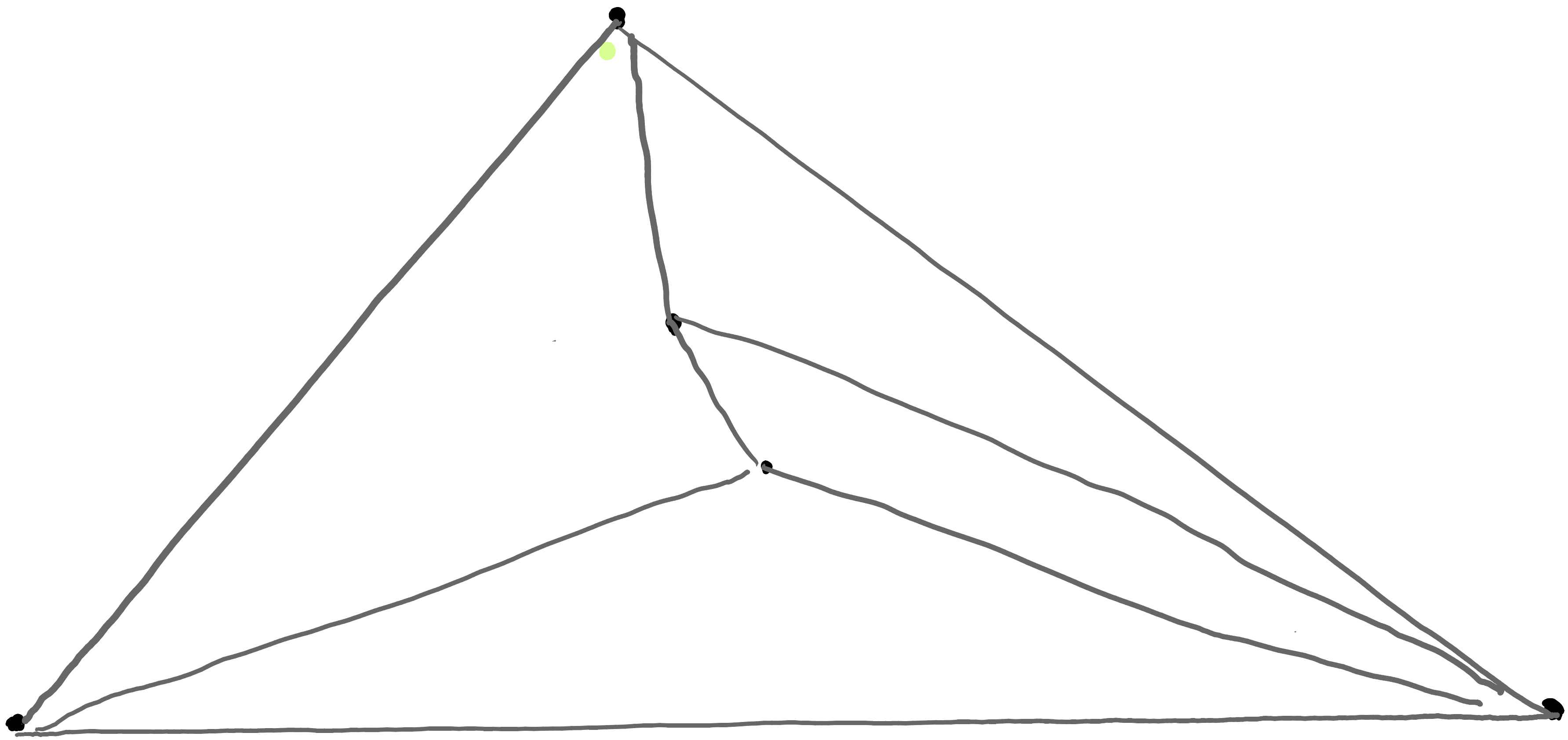




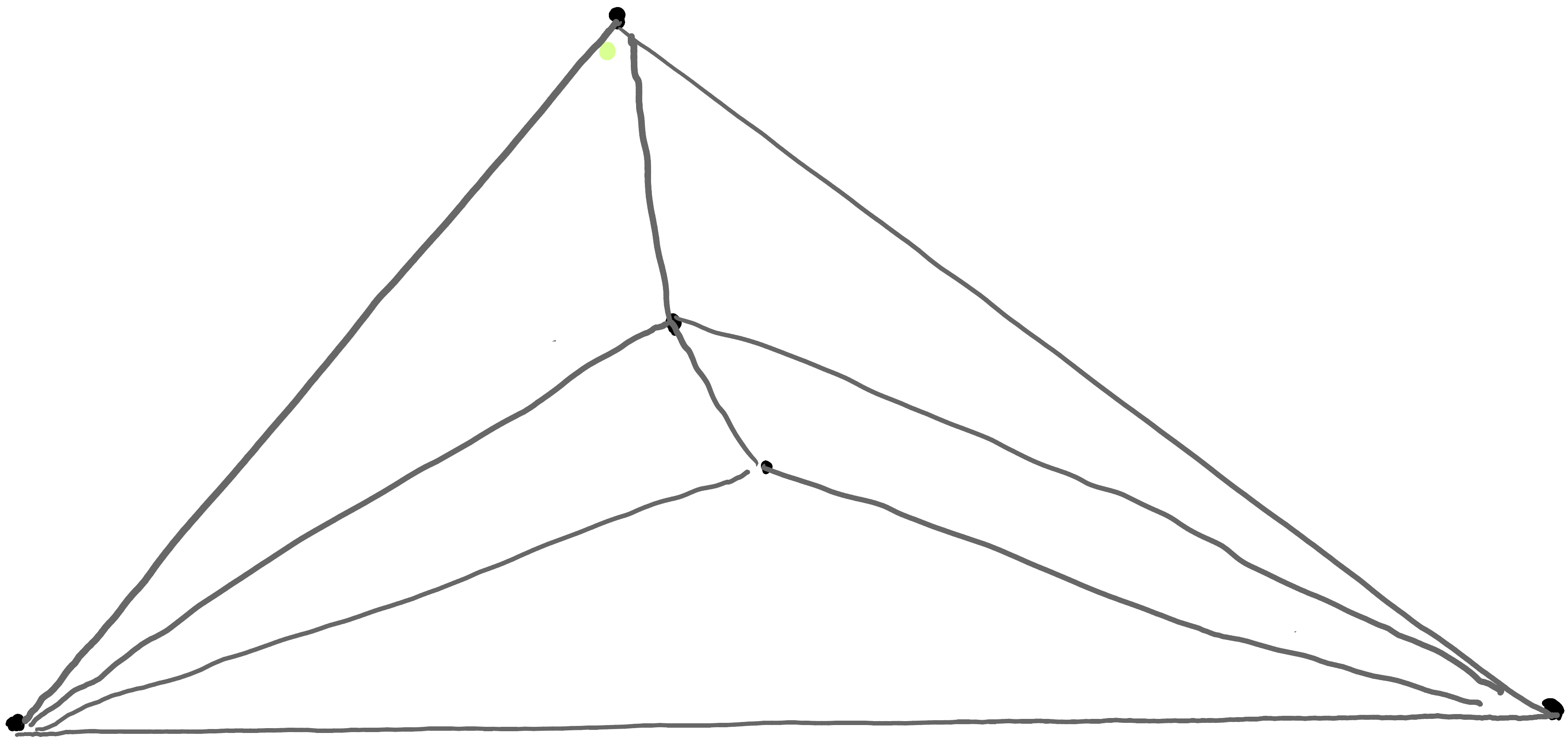
Level 5

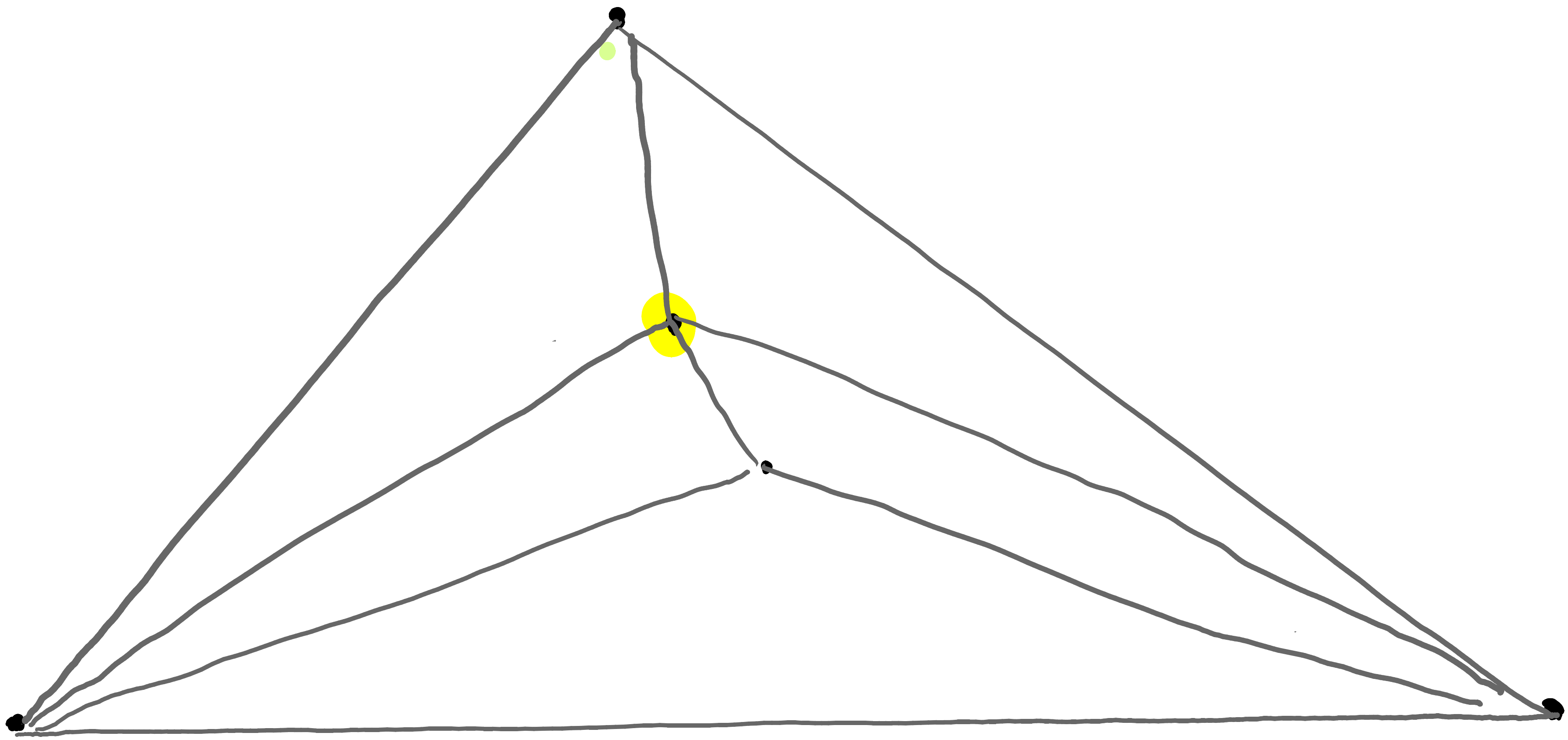


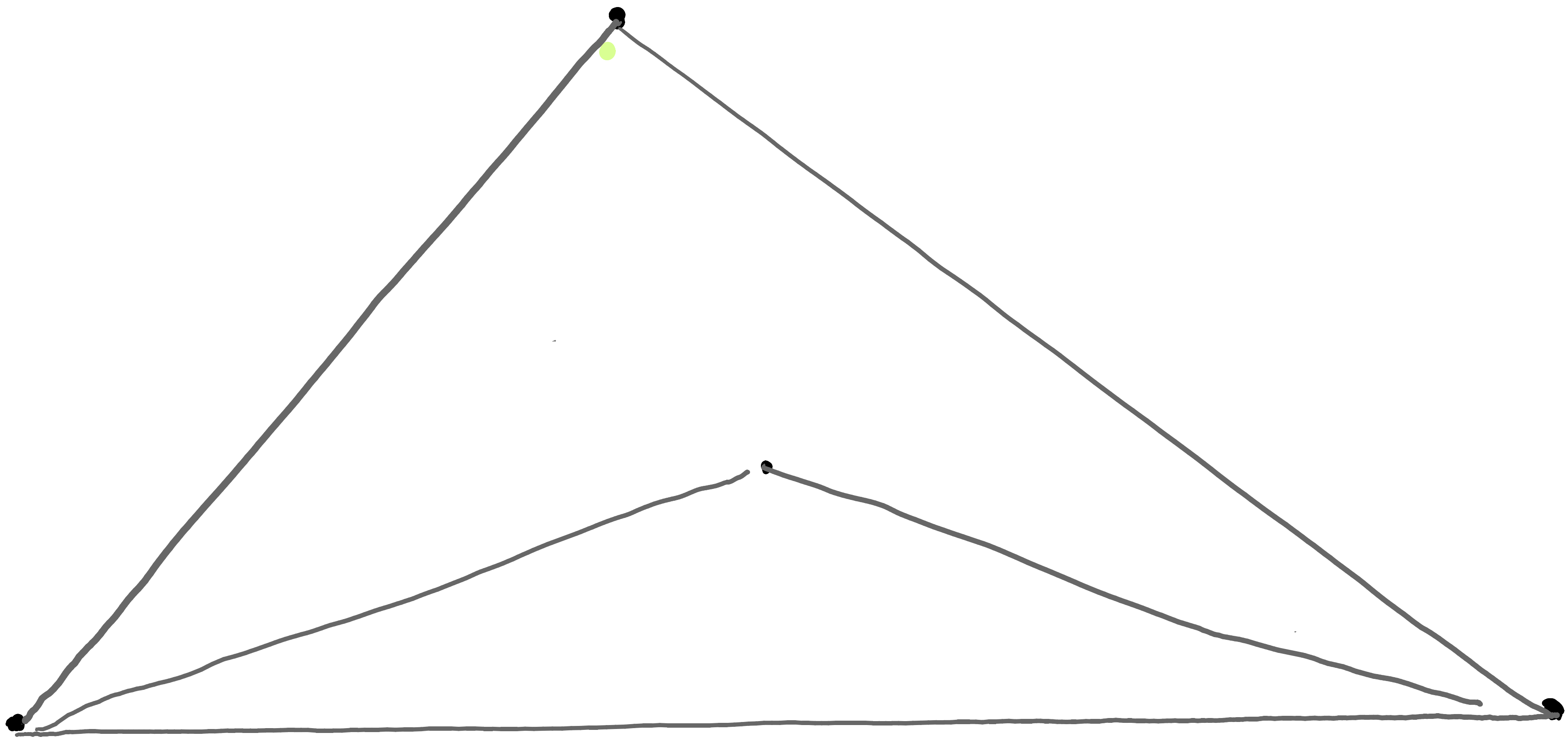




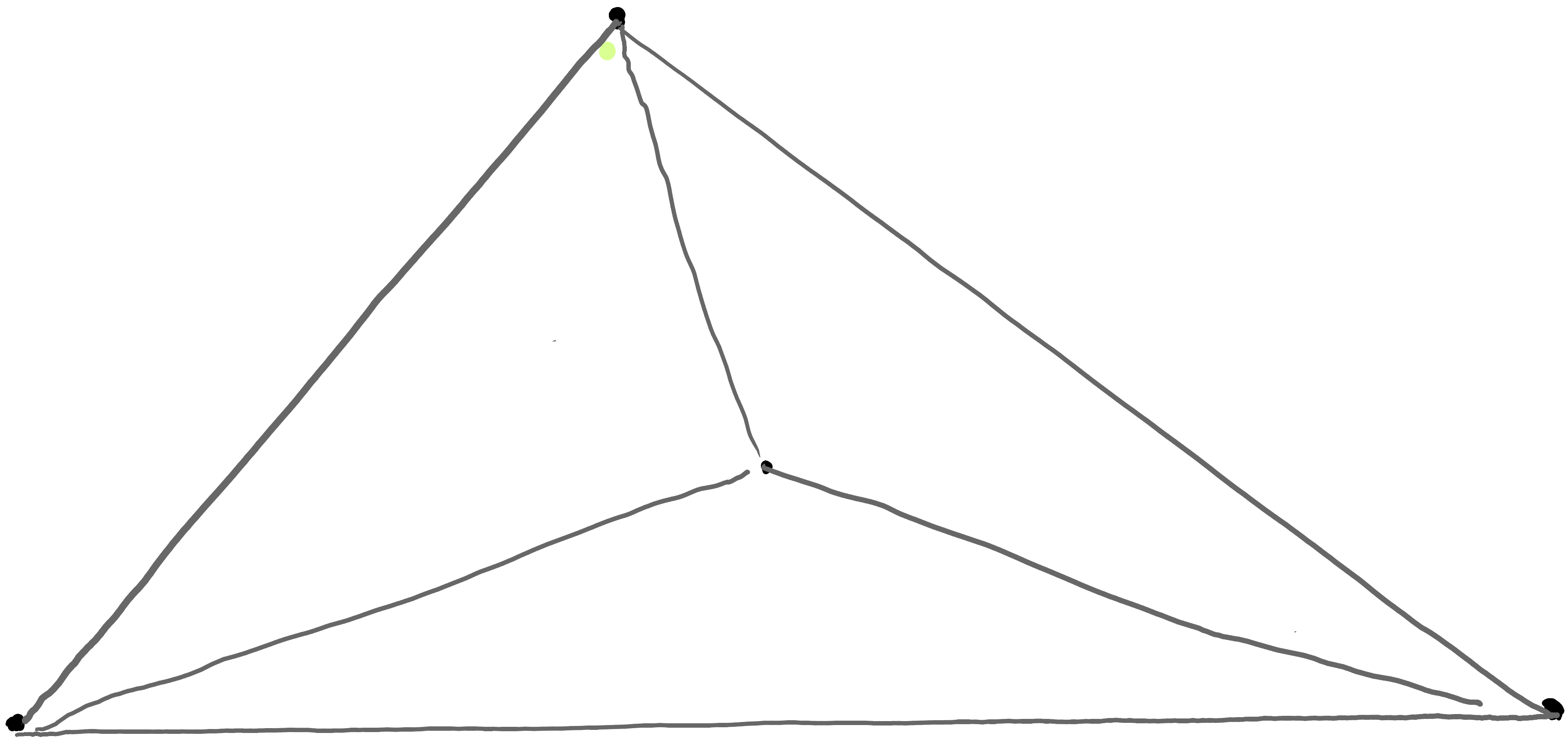
Level 6

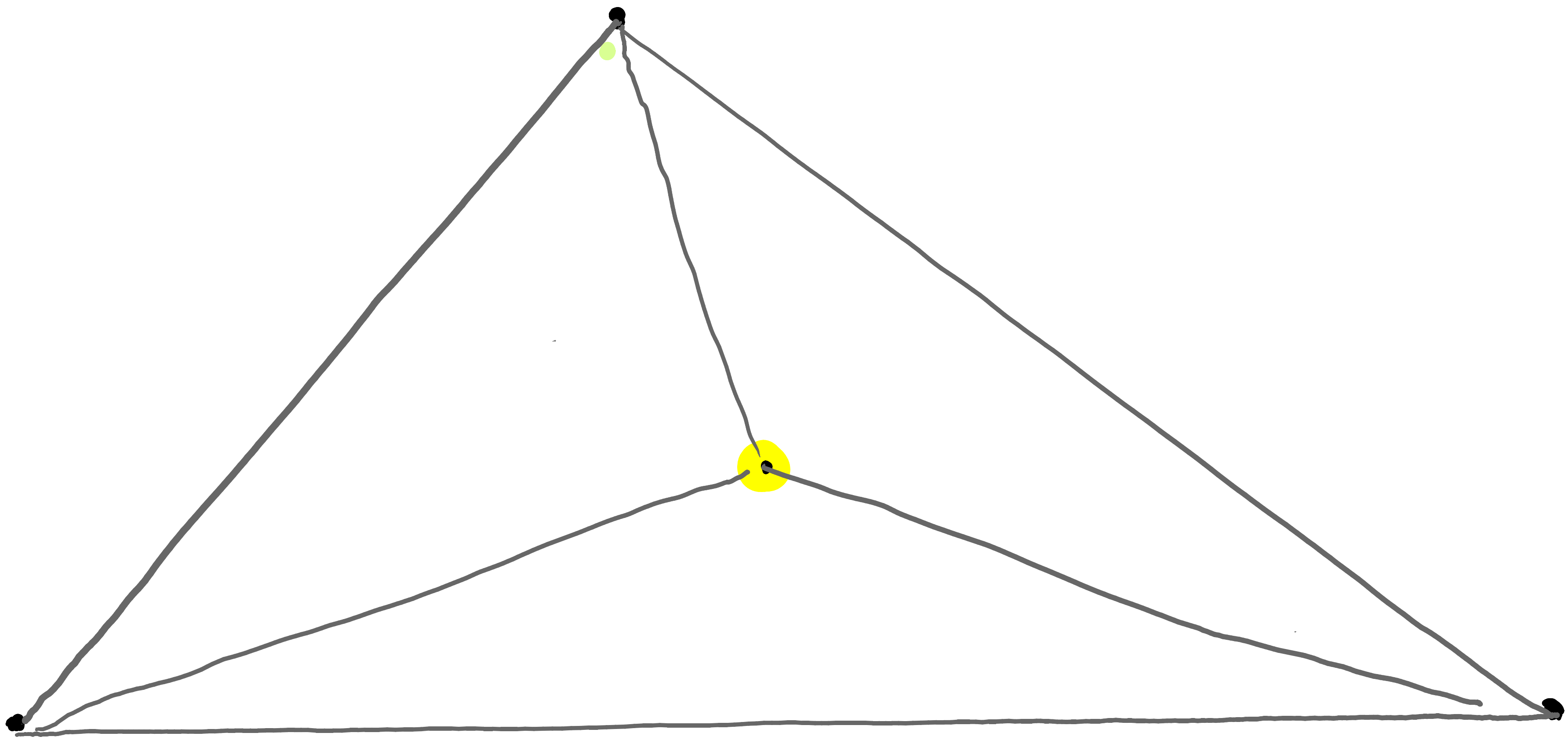




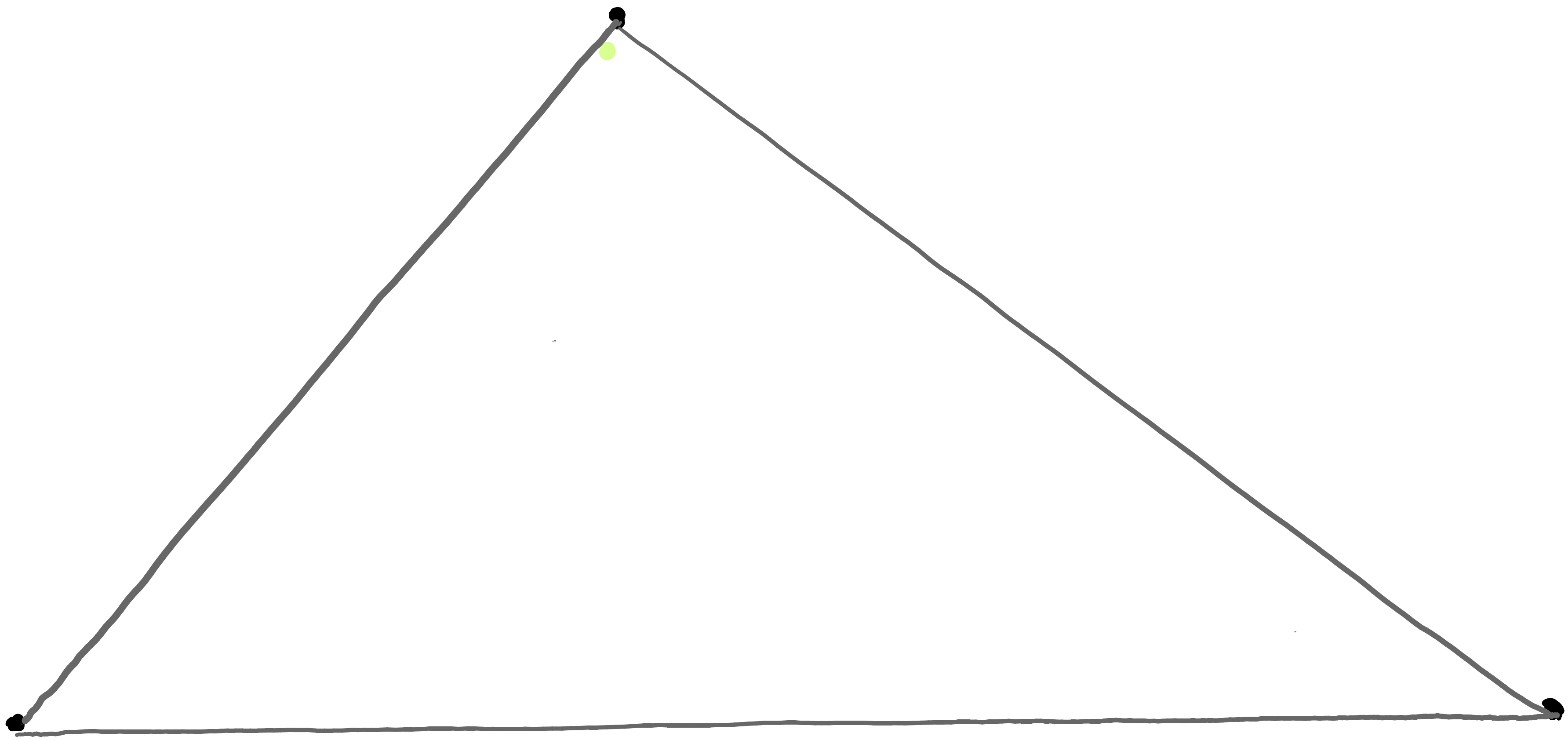


Level 7





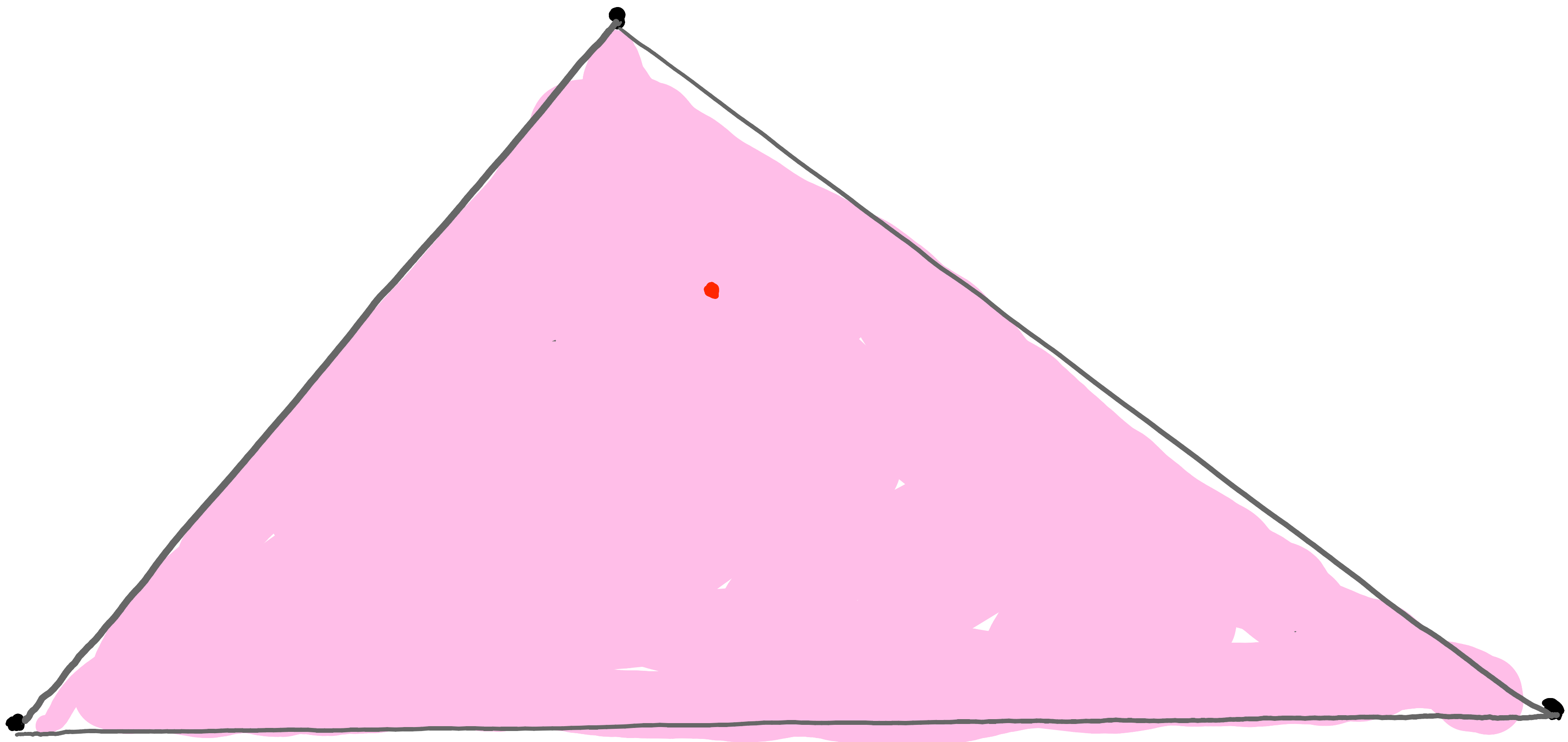
Level 8



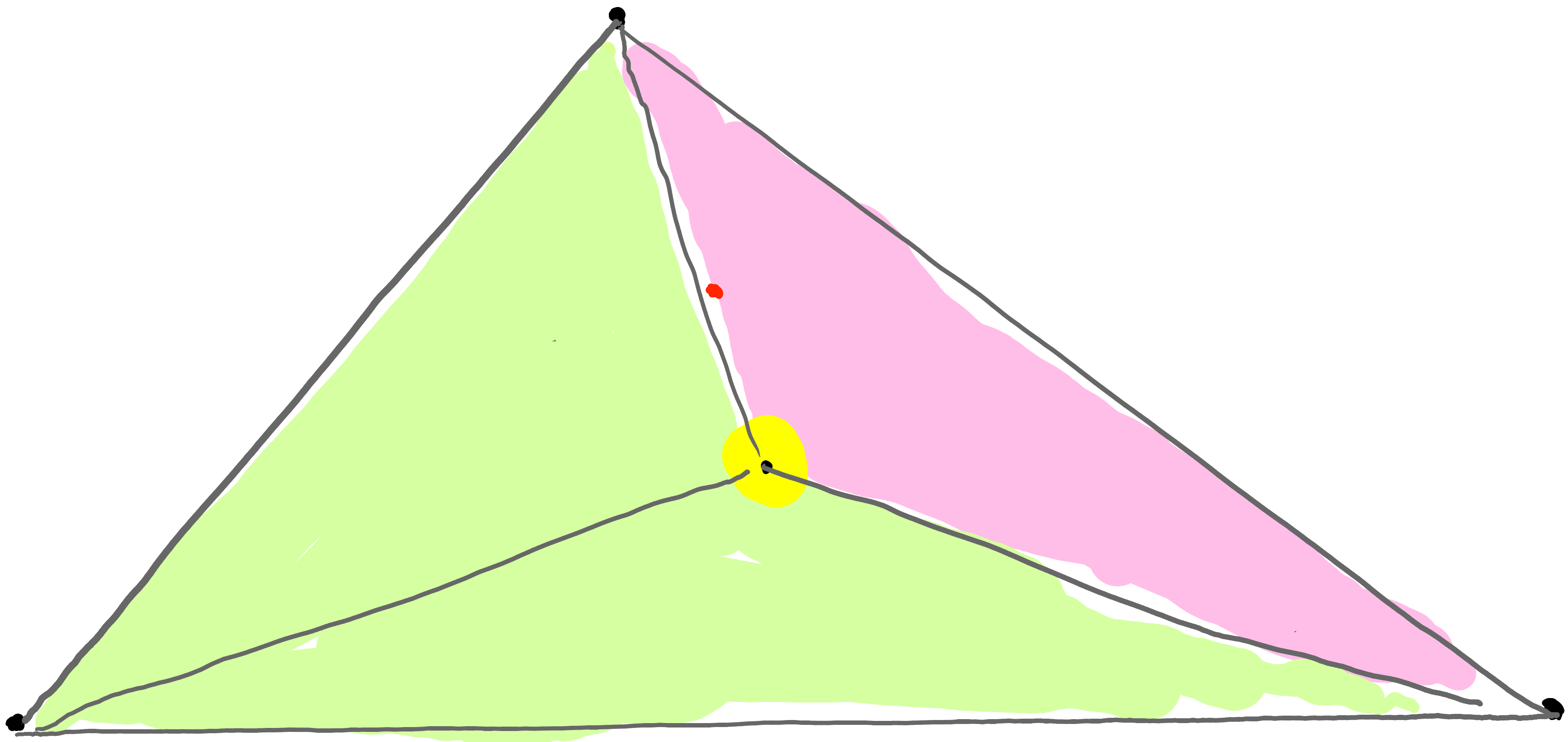
How many levels?

$$\log_{\frac{15}{16}} n \approx 10.7 \log_2 n$$

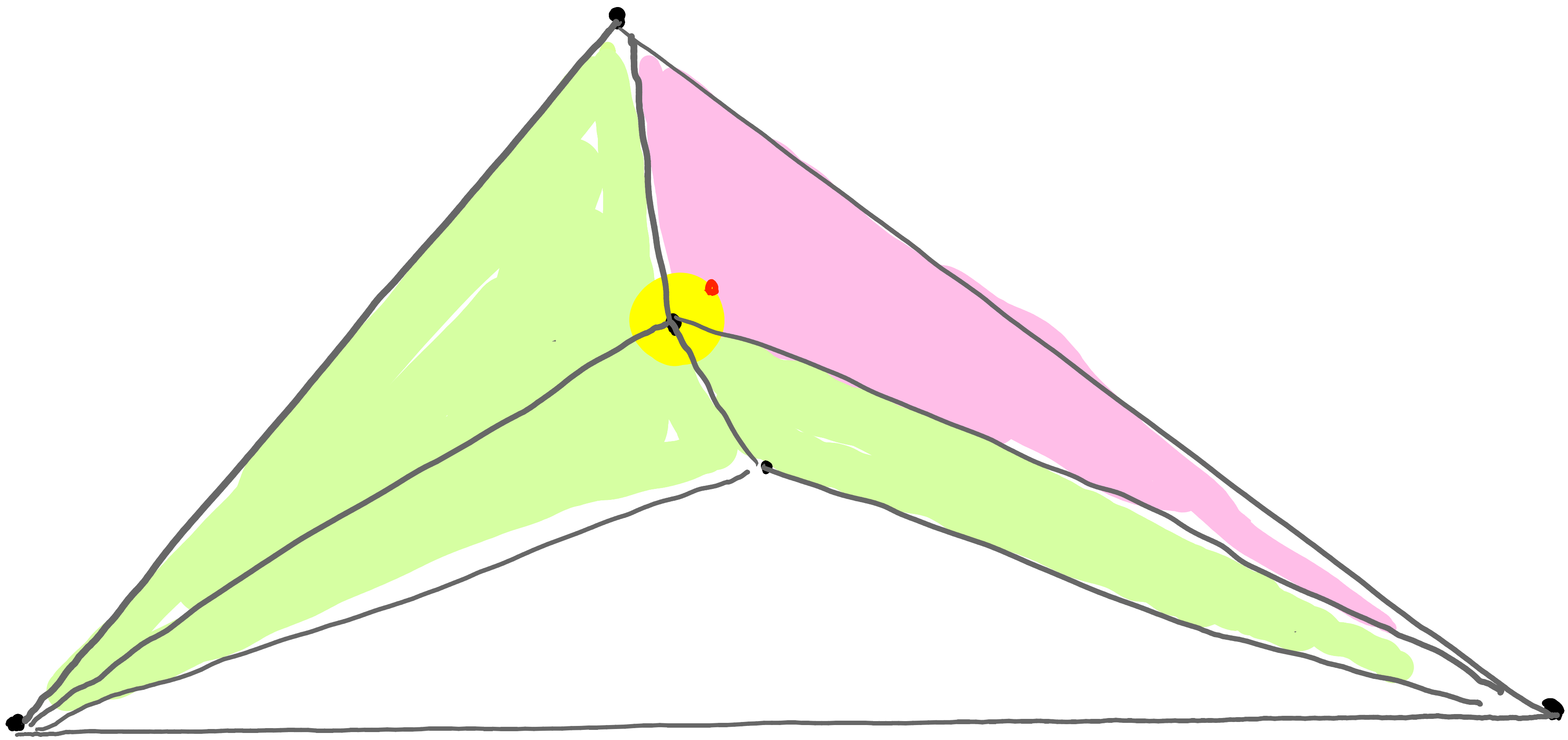
Level 8



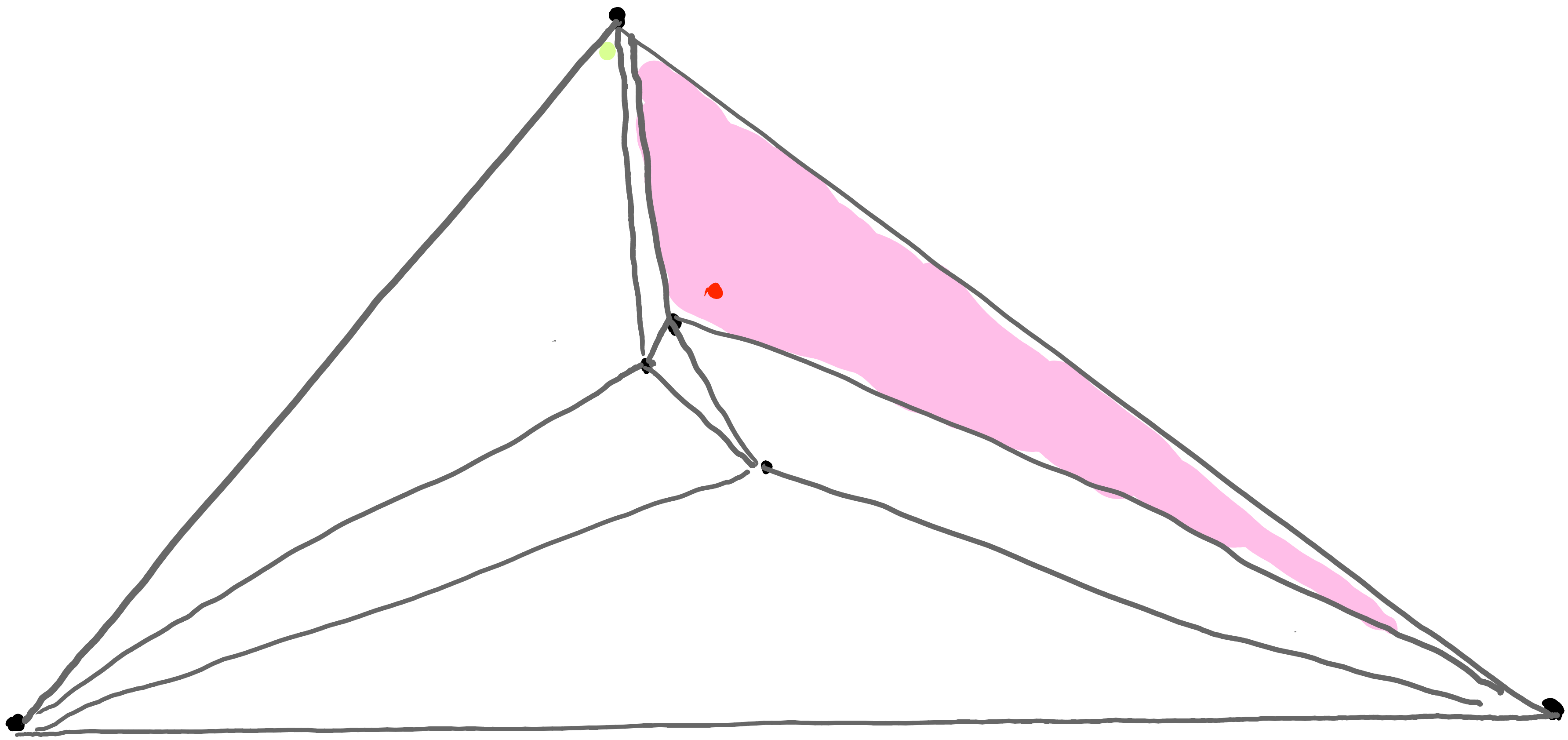
Level 7



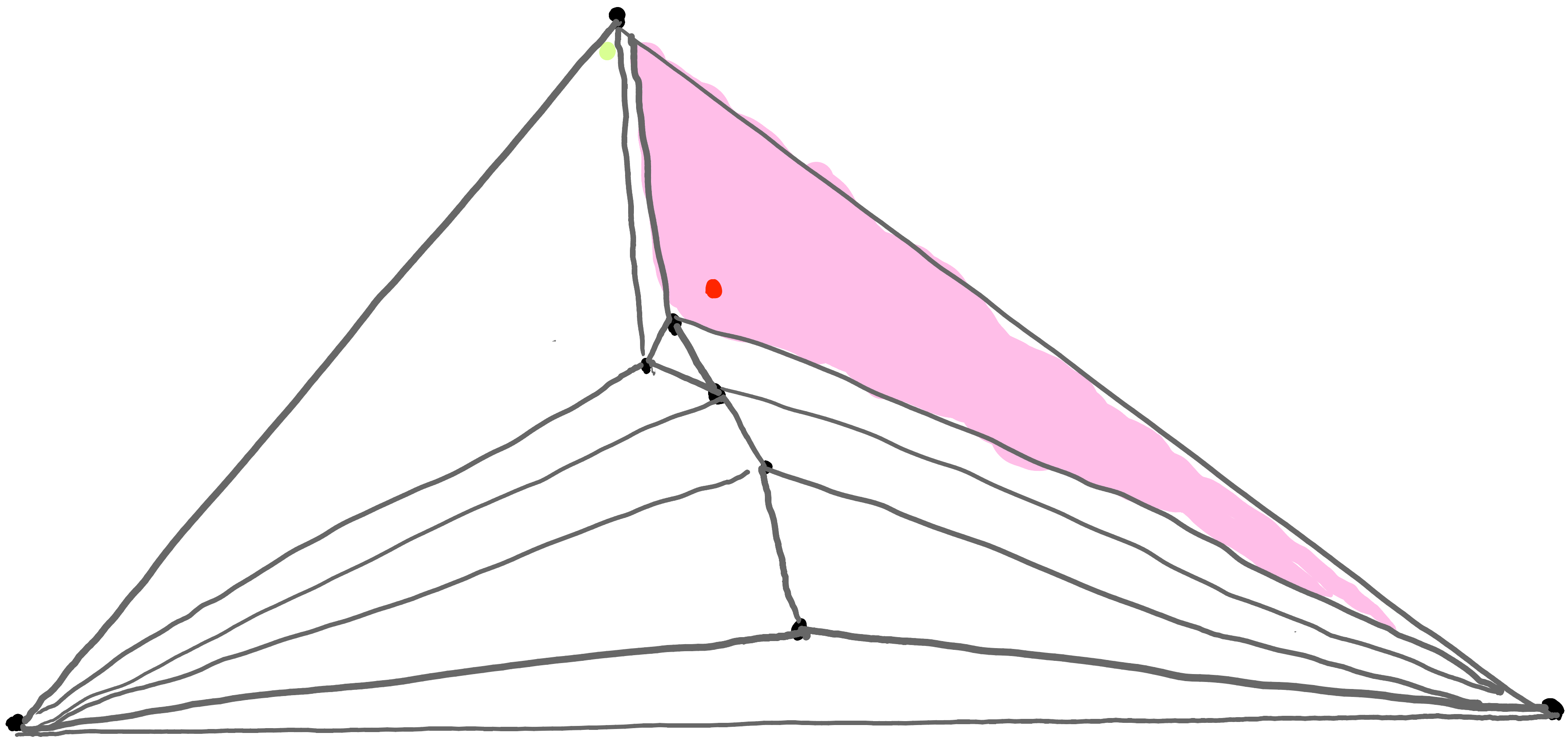
Level 6



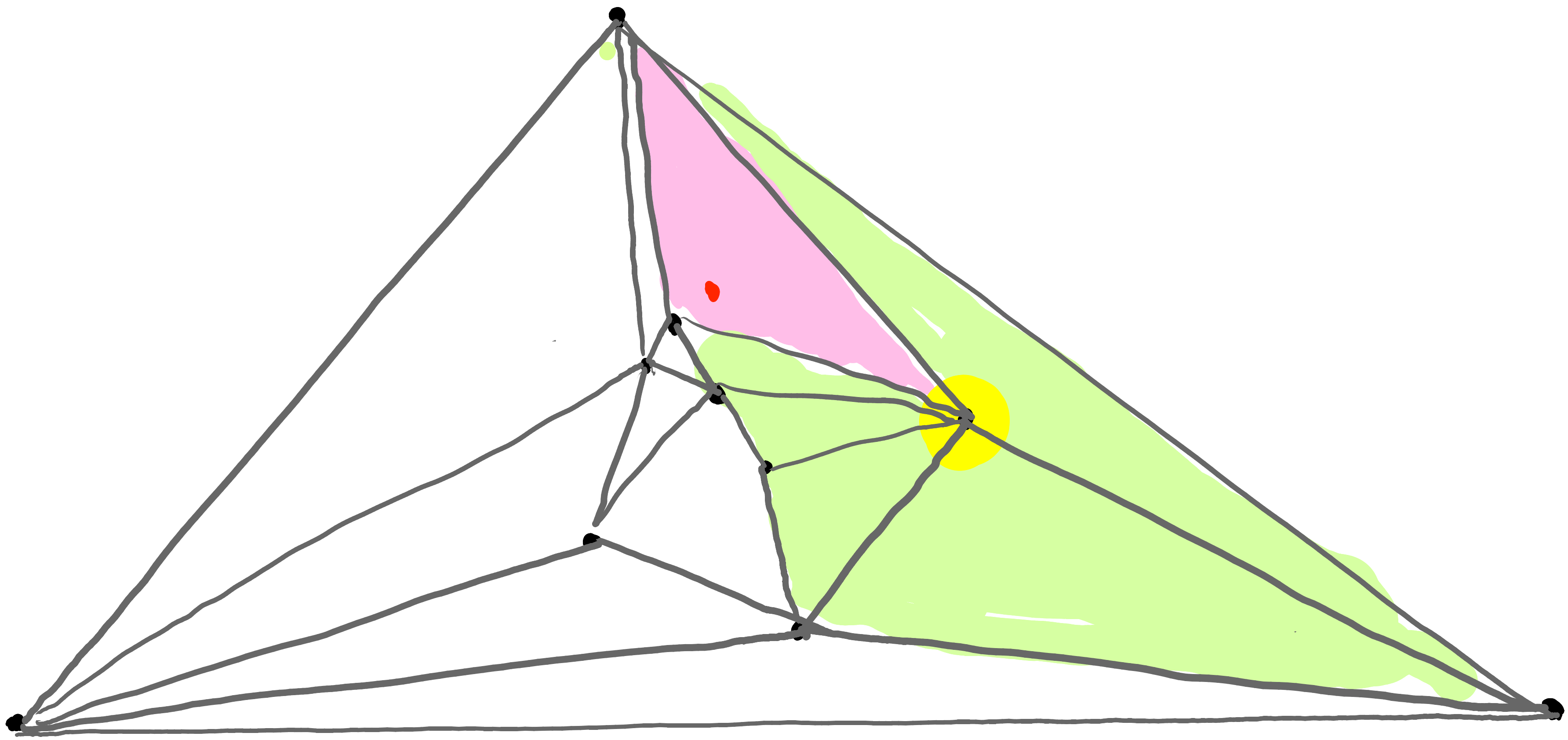
Level 5



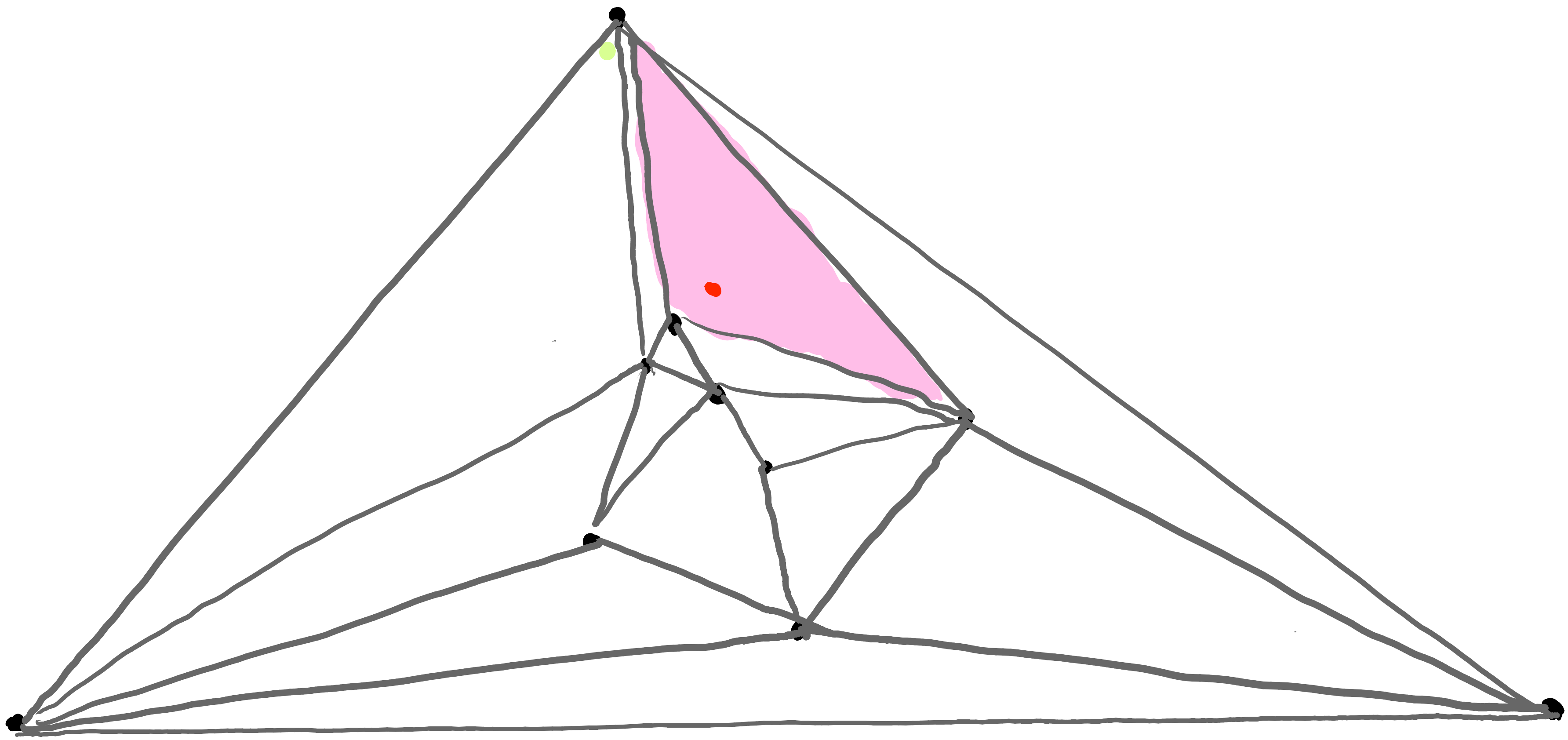
Level 4



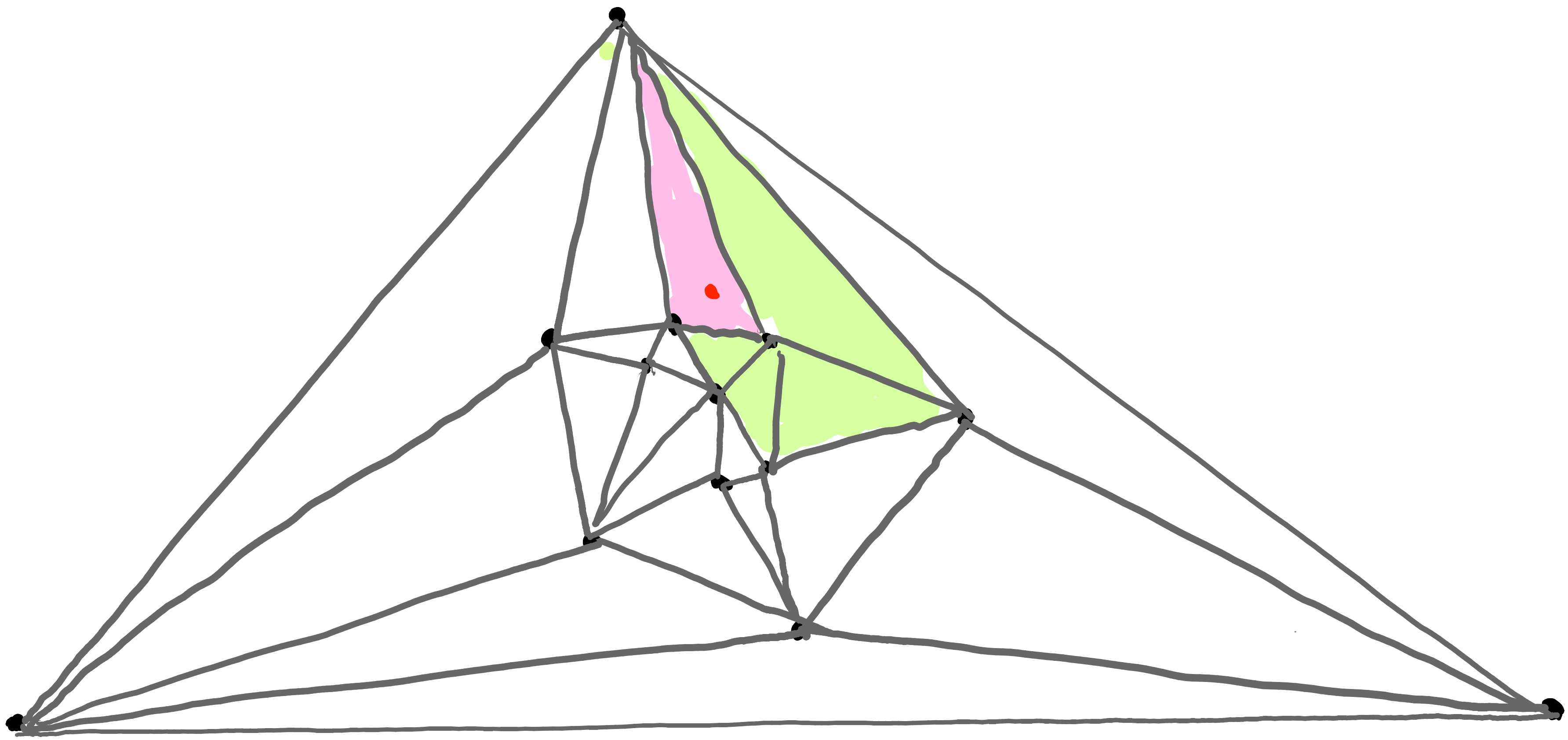
Level 3



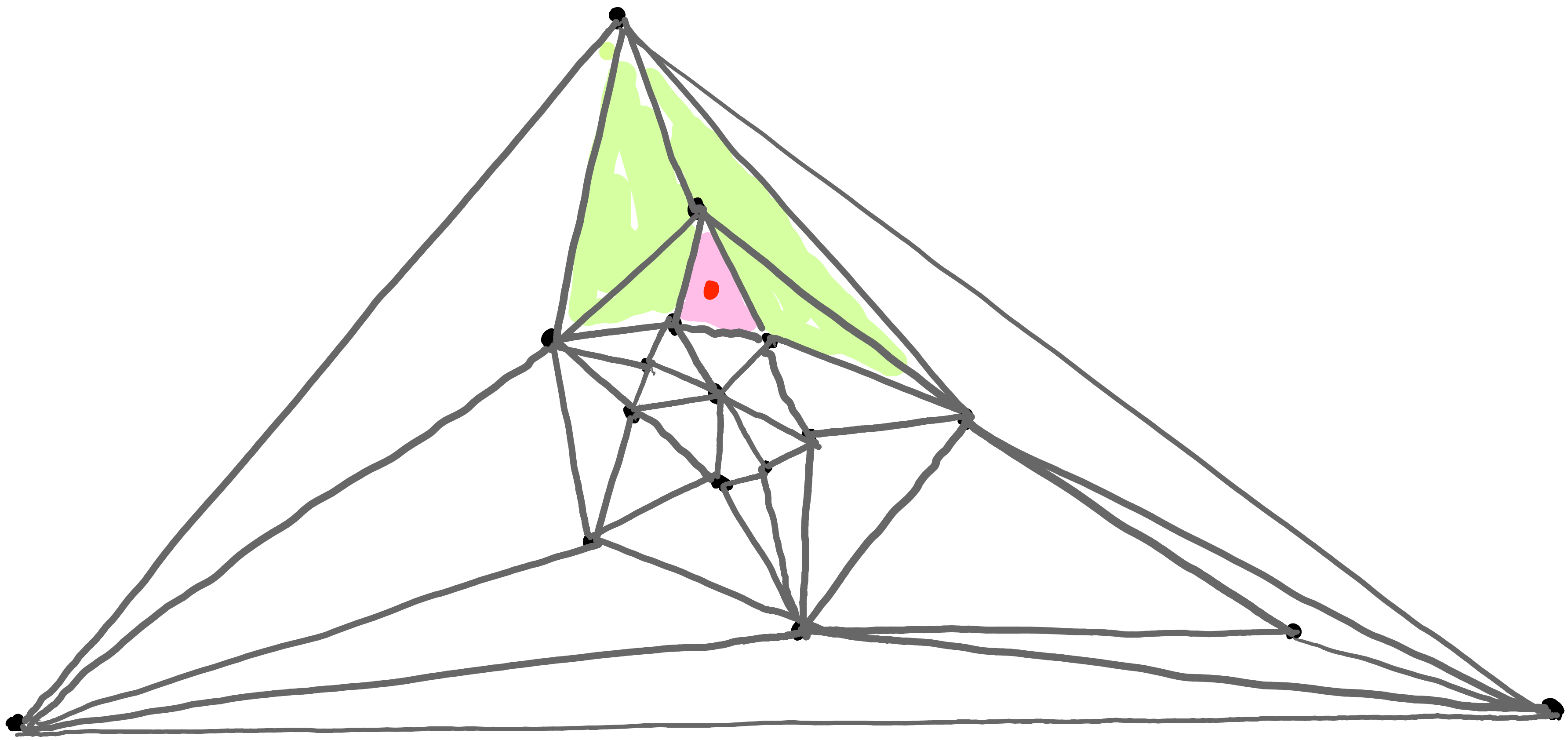
Level 3



Level 2

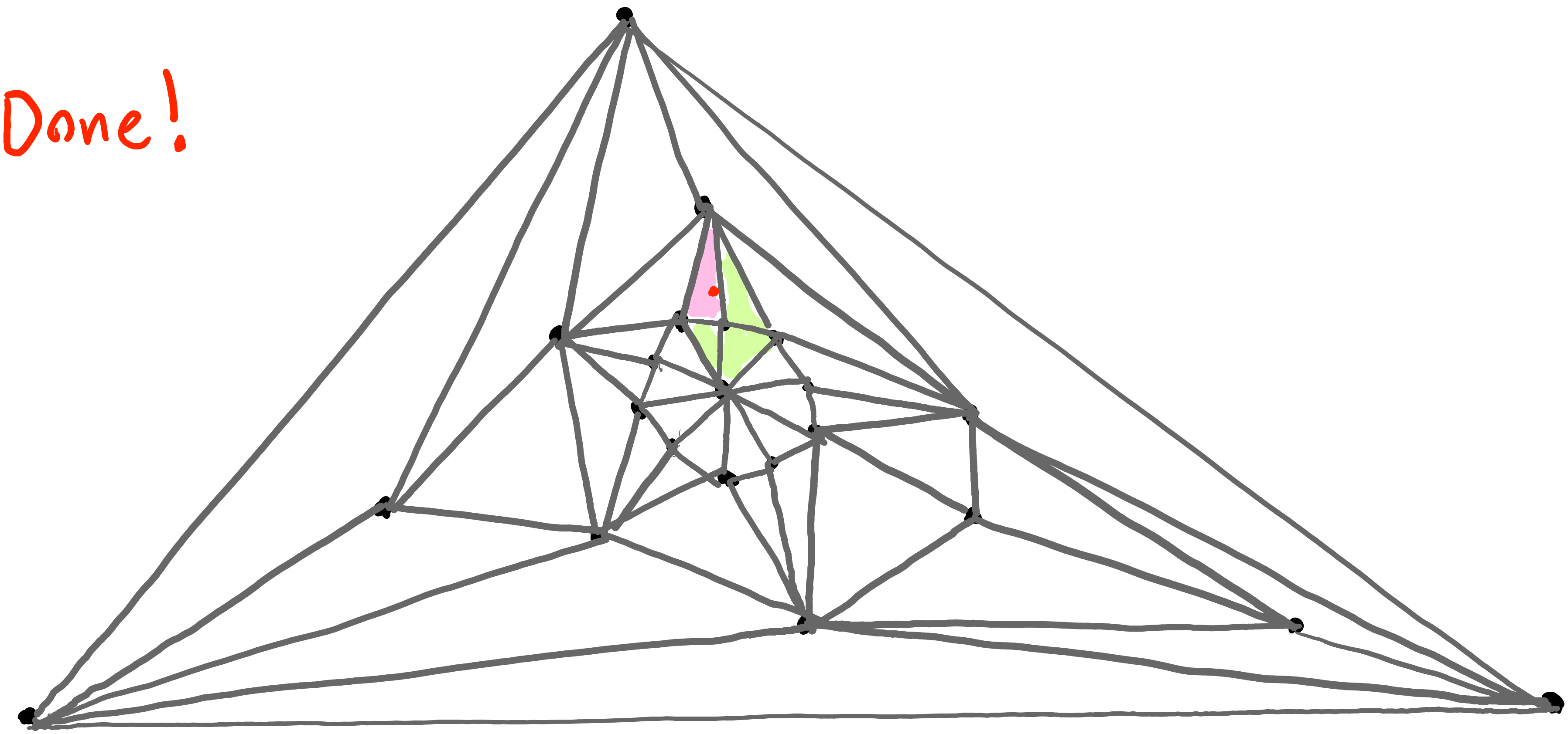


Level 1



Level 0

Done!



Summary

Summary

+ Construction time $O(n)$

Summary

+ Construction time $O(n)$

+ search $O(\log n)$

Summary

+ Construction time $O(n)$
+ Search $O(\log n)$
+ Deterministic

Summary

+ Construction time $O(n)$
+ Search $O(\log n)$
+ Deterministic
+ Only method with this combination

Summary

+ Construction time $O(n)$
+ Search $O(\log n)$
+ Deterministic
+ Only method with this combination
− Constants terrible

Summary

+ Construction time $O(n)$
+ Search $O(\log n)$
+ Deterministic
+ Only method with this combination
− Constants terrible
− Requires triangulation

Summary

+ Construction time $O(n)$
+ Search $O(\log n)$
+ Deterministic
+ Only method with this combination
− Constants terrible
− Requires triangulation
+ Can triangulate any simple polygon in $O(n)$ time [Chazelle]

Summary

+ Construction time $O(n)$
+ Search $O(\log n)$
+ Deterministic
+ Only method with this combination
- Constants terrible
- Requires triangulation
+ Can triangulate any simple polygon in $O(n)$ time [Chazelle]
- Chazelle's method is difficult

Summary

+ Construction time $O(n)$

+ Search $O(\log n)$

+ Deterministic

+ Only method with this combination

− Constants terrible

− Requires triangulation

+ Can triangulate any simple polygon in $O(n)$ time [Chazelle]

− Chazelle's method is difficult

− No obvious way to make dynamic