1 The Graph Drawing Symposium and Contest

For over 25 years, an international symposium of Graph Drawing and Network Visualization takes place annually. Wo überall bisher? In the year 2021, the 28th International Symposium of Graph Drawing and Network Visualization will be held from September 15th to 17th in Tübingen.
Part of the symposium is a traditional Graph Drawing Contest. The contest consists of two parts - the *Creative Topics* and the *Live Challenge*. The main focus for the Creative Topics lies on the creation of drawings of two given graphs. Aspects to consider for the visualization are clarity, aesthetic appeal and readability.
On the other hand, the Live Challenge is held similar to a programming contest. Participants, usually teams, will get a theme and a set of graphs and will have one hour of processing. The results will be ranked and the team with the highest score wins the competition. The teams will be allowed to use any combination of software and human interaction systems in order to produce the best results. Usually, the challenge is derived from a theoretical optimization problem.
This year, there will be two categories that are judged independently:

Automatic: The teams will use their own defined toolchain. Therefore, the challenge graphs will be large ones.

Manual: A given graph tool will only provide the functionality to manually move objects. This prohibits any algorithm to work on the solution.

1.1 The challenge

There has been recent attention to the edge-length ratio of a planar drawing, which describes the ratio between the lengths of the longest edge and the shortest edge in a drawing.
This year, the main topic addresses an optimization problem, namely the minimization of the edge-length ratio of poly-line drawings of planar, undirected graphs on a fixed grid. For a poly-line edge, the edge-length is the sum of the line segment lengths.

The input consists of a JSON file with the following entries:

nodes Every node has an unique ID value between 0 and the amount of nodes - 1, a value for the x and y coordinate each, delimited by the width and height

edges Every edge has an ID for source and destination each and an optional list of bend points, specified in x and y coordinate

width (optional) The maximum x-coordinate of the grid. If unspecified, the width is set to 1,000,000.

height (optional) The maximum y coordinate of the grid. If unspecified, the height is set to 1,000,000.

bends The maximum number of bends allowed per edge

The results of the optimization are also JSON files. The planarity of the graph shall be preserved and the poly-line edge-length ratio minimized by relocation of the nodes.

For the teams participating with their own tools, an embedding might not be given with the input. For the participants working manually, an embedding is already given beforehand.

2 1. May 2021 - Report (a summary)

In order to investigate the edge-length ratio, the first approach was to draw simple example graphs and simply look for the edge-length ratio behaviour.

2.1 First example - One triangle

Lemma 1. *There exists a graph which is not drawable on a fixed grid with an edge-length ratio of 1.*

Proof. Consider a triangle. In order to draw it with equally long edge lengths l, the height must value $\frac{\sqrt{3}}{2}l$.

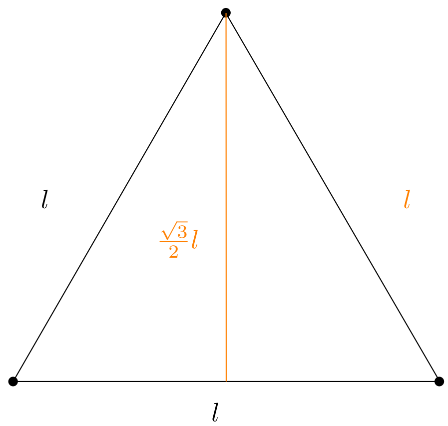


Figure 1: Triangle

Since $\sqrt{3}$ is an irrational number, therefore there exists no combination of coordinates on a grid in order to represent the triangle. Otherwise, there would exists two integers to represent $\sqrt{3}$ as a fraction. Therefore, there exists no drawing of a triangle with edge-length ratio 1. □

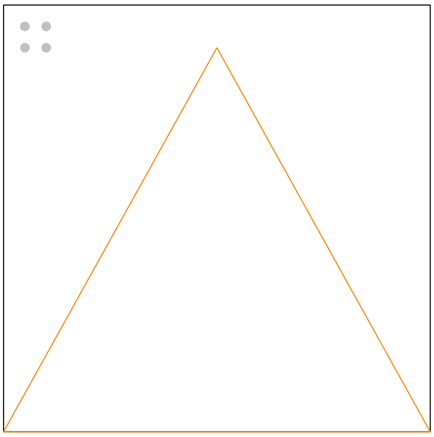
If we want to get close to an egde-length ratio of 1, we are bound to approximate the cofactor $\sqrt{3}/2$ in the height of the triangle.

Observation 1. *W.l.o.g., let the given grid be of size $g \times g, g \in \mathbb{N}$. For c, it holds:*

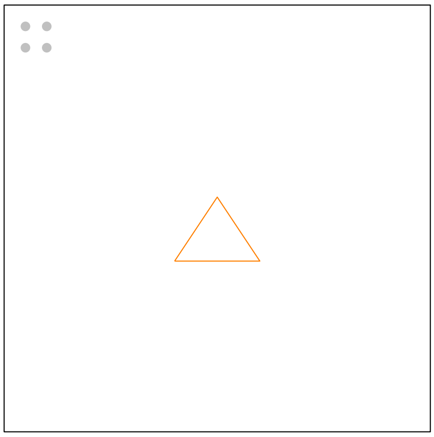
$$c = 0,8660254 \approx \frac{\sqrt{3}}{2}$$

Then, it is better for the edge-length ratio to draw the triangle as large as possible on the grid so that the approximation gets more precise.

So one first intuition is to use all the given area on the grid.



Triangle A

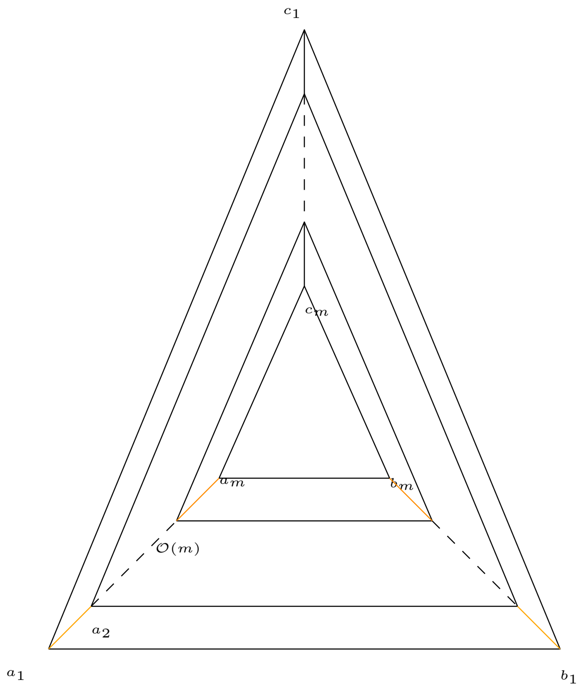


Triangle B

$$\text{ratio}_A < \text{ratio}_B$$

2.2 Second example - m Triangles, nested and triconnected

Take a look at the following drawing:

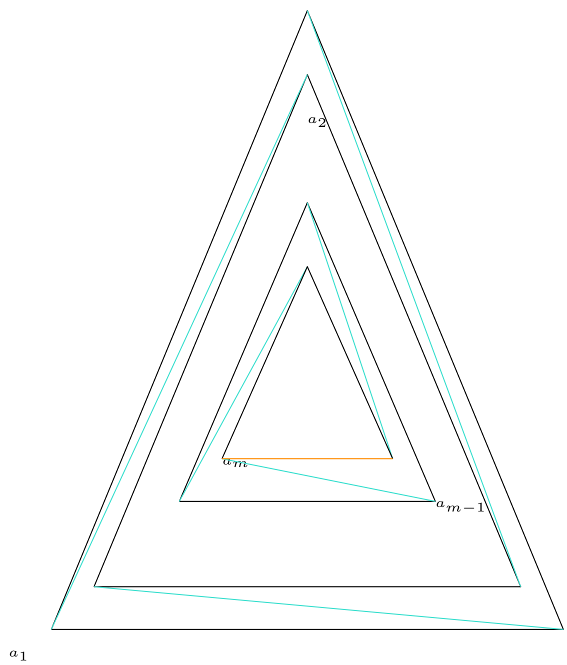


Drawing Γ of m nested triangles

The i-th triangle is defined as the connected set of vertices $\{a_i, b_i, c_i\}$. One difficulty of a resizing is the nestedness of the triangles. On one hand, if an inner triangle gets larger, then the whole drawing might get larger, meaning that the longest edge got longer. On the other hand, if the triangles distances differ by a constant, the diagonal, orange-colored edge stay small.

Observation 2. *The solution is to reposition the vertices of every triangle. One can imagine, that each inner triangle is "flipped" by one more turn. No bends were used yet.*

Then, the following new drawing is derived by this approach.

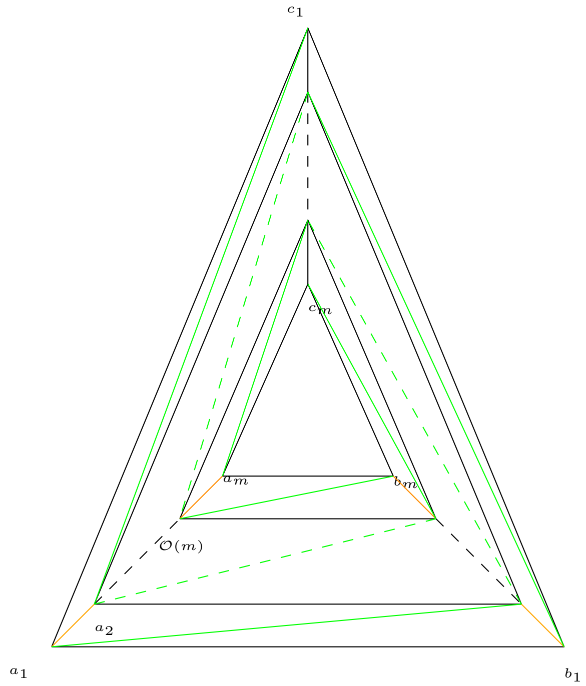


New drawing Γ' of m nested triangles

With help of this vertex repositioning, the shortest edge now is a longer one - still illustrated in orange. When the shortest and the longest edge are within one of the m triangles and the distance between each triangle is constant, then the ratio lies in $\mathcal{O}(n)$ and is independent of the grid size.

2.3 Variant of the second example - more edges!

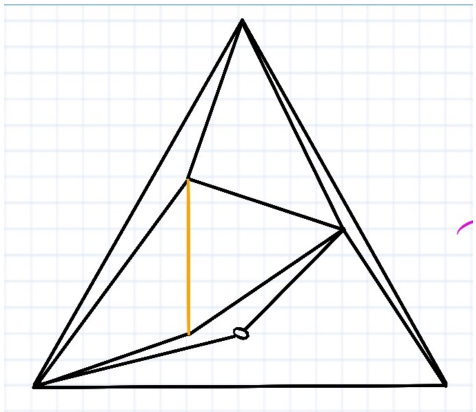
The previous example has got more edges in a way, that each vertex is of degree at least 4. The following drawing illustrates the altered graph:



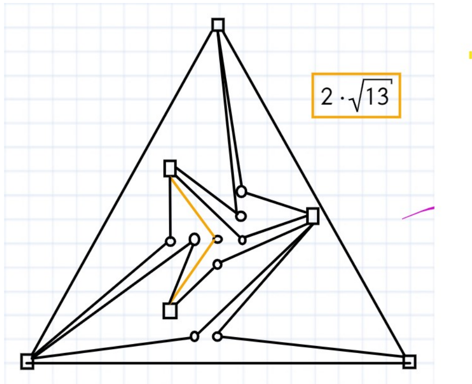
Drawing of altered previous graph, new edges are illustrated with a green color

Now, the “turn“ of the triangles is restricted since we might violate the property of planarity. One question was, how the triangles should be drawn generally. So, in order to get an idea, the drawing for $m = 2$ was optimized until there is nothing left to do.

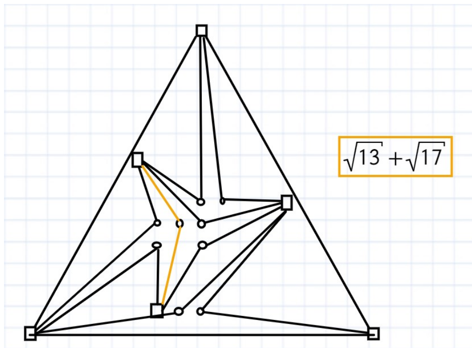
Observation 3. *By “keeping the triangles close to each other“, it is possible to further reduce the edge-length ratio.*



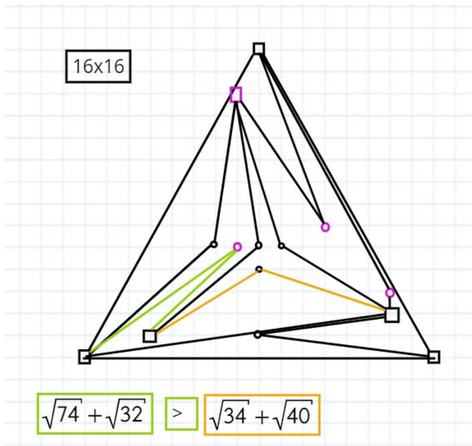
(a) Start the optimization here. The shortest edge is illustrated with an orange color



(b) Bends were included. The Unit Length equals the length of one box.



(c) Vertex and bend repositioned for a better result



(d) After greedily elongate the shortest edge in a couple of iterations, it gets clear, how the triangles could be drawn with a satisfying ratio.

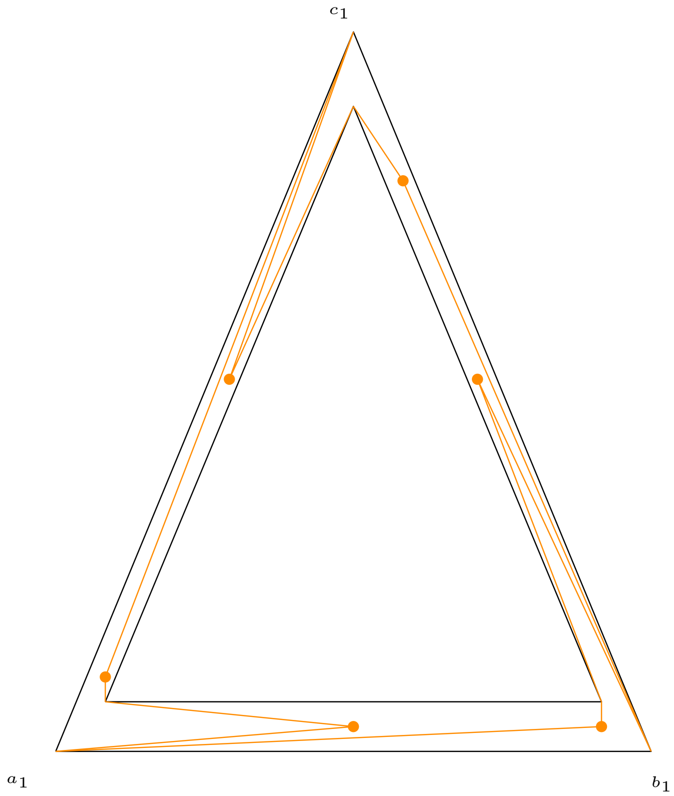
This leads to the following approach, the discussion to the steps follow further below.

Algorithm 0: Algorithm sketch for a drawing of the example graph

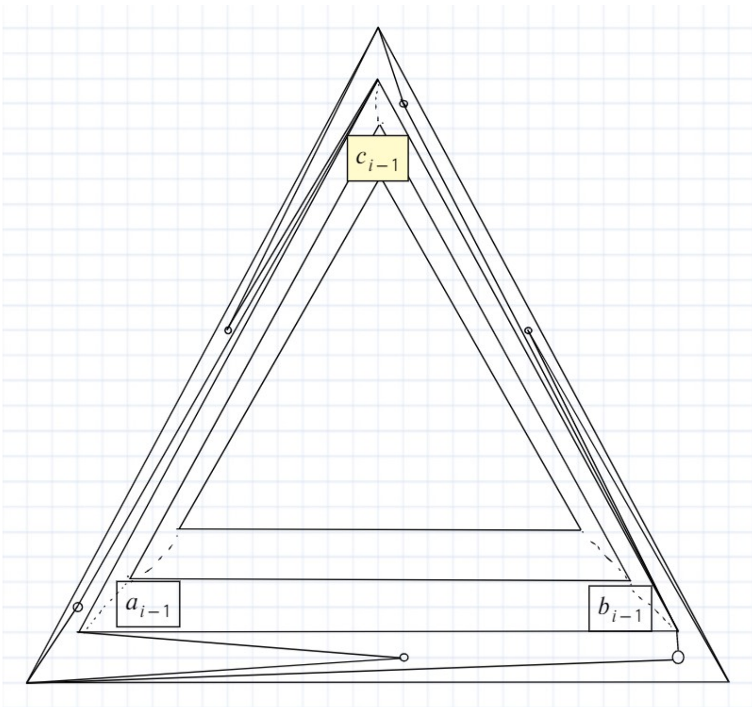
```
Input: Graph consisting of m triangles connected as illustrated above,
       Gridsize, one bend allowed
Output: Drawing with a satisfying edge-length ratio
Draw the outermost triangle as big as possible on the grid with the
  approximated height
for i = 2 to m do
    Place the i-th triangle inside the predecessor by a constant c
    Place bend points between those triangles and draw the edges with length
      bound by the shortest edge of the inner triangle and the longest edge of the
      outer triangle
```

Details:

1. Recall the approximation for a uni-sized triangle of the first section for the triangle.
2. It is important to have grid points between two consecutive triangle drawings. The bends must be placed somewhere.
3. To get an idea, how exactly the bend points are set, please, take a look at the following picture.



(a) This is the output of the algorithm with $m = 2$.



(b) In this case, deg(a_1)=5, deg(b_1) = 3, deg(c_1) = 4

The bend point for w.l.o.g. (a_{i-1}, a_i) lies centralized between $a_{i-1}, b_{i-1}, a_i, b_i$. Analogue for the other ones. The bend point for w.l.o.g. $(a_{i-1}, b_i$ lies below b. To guarantee the existence of those points, it is important not to choose the "shrinking constant" c too small.

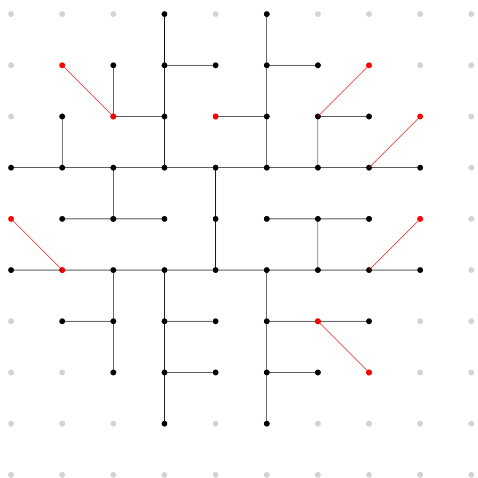
Lemma 2. *With one bend allowed per edge, the graph above, inheriting m triangles, will have an edge-length ratio in $\mathcal{O}(n)$.*

Proof. If the construction of the drawing is correct, then the innermost triangle t_m will inherit the shortest edge and the outerface t_1 will contain the longest edge. Since the triangles have a constant distance pairwise, the differences between the edge lengths of the outermost and the innermost triangle is linear to the amount of triangles. Therefore, the ratio between the longest edge on the outermost triangle and the shortest edge on the innermost triangle grows linear with the amount of triangles. Hence, the ratio lies in $\mathcal{O}(n)$. □

2.4 Third example - a complete binary tree with depth d

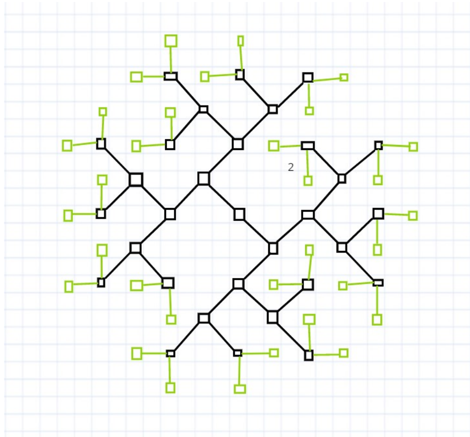
A binary tree is 1-connected. It is of interest, how the edge-length ratio behaves for a given depth d. In order to get a feeling, drawings of uniform long edges were created and it was observed, at which depth it is not possible to place further vertices.

Observation 4. *When the drawing starts with horizontal or vertical edges, it is not possible to draw a complete binary tree of depth 5 with an edge-length ratio of 1.*



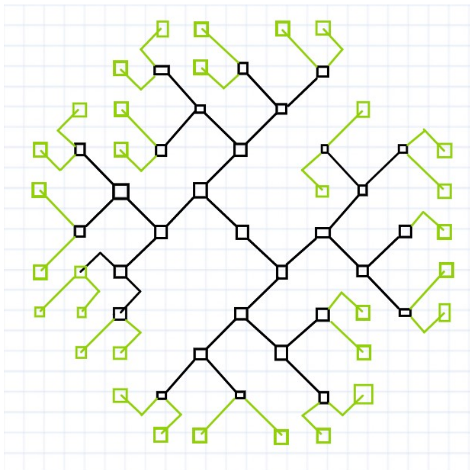
At some point, there is simply not enough area for 2^d vertices

Observation 5. *If starting diagonally first, a complete binary tree of depth 5 is drawable on a 11x11 grid with a ratio of $\sqrt{2}$.*



It seems that, when starting diagonally, there might be more grid points availiable

Lemma 3. *Allowing one bend, it is possible to draw a complete binary tree of depth* 5 *with an edge-length ratio of 1.*

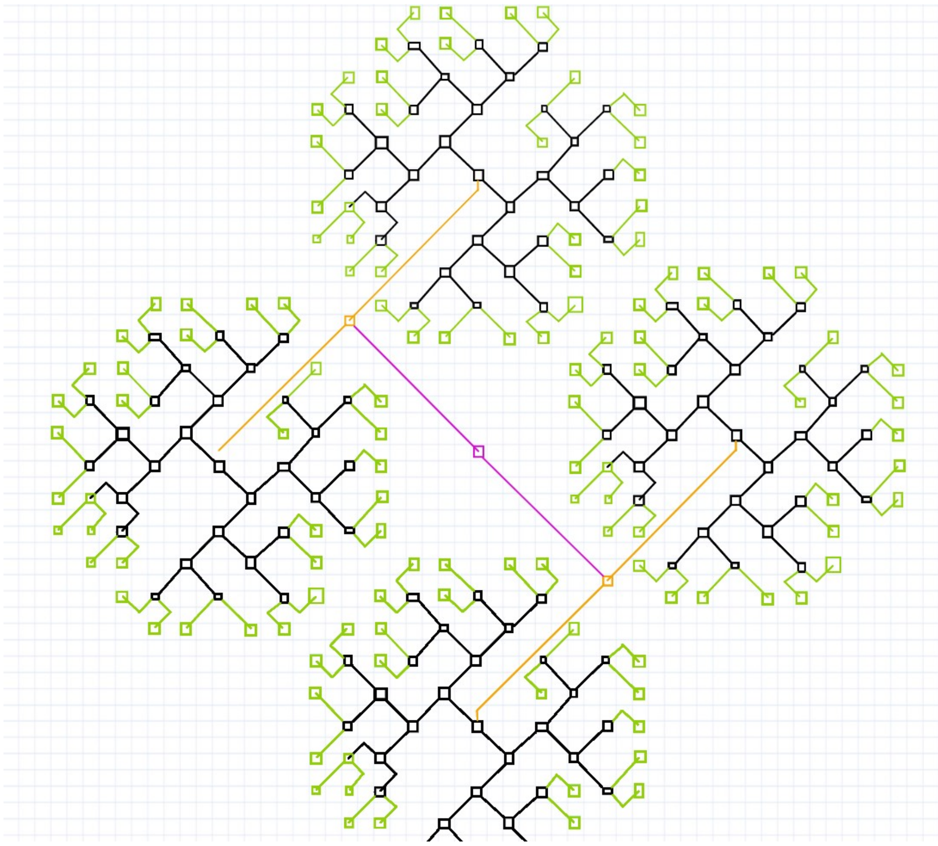


Using bends, all edges are of the same length

Proof. □

Theorem 1. *A complete binary tree with depth d can be drawn with an edge-length ratio of $\mathcal{O}(\sqrt{n})$*

Sketch of a proof. As already seen, for the depth up to 5, the edge-length ratio values 1. Incrementally, the drawing of i-th depth is achieved by copying the drawing of $i-1$ and each roots of the subtrees with a vertex. When looking at the drawing of depth 7, then we observe that the ratio is the same as of the drawing with depth 6.



Depth 6 in orange, depth 7 in purple

The ratio can be expressed as:

$$\text{ratio}_d = 2^{\lfloor \frac{d}{2} \rfloor - 1} + \frac{1}{2} \in \mathcal{O}(n)$$

□

2.5 General observations / thoughts

- Draw large, thinking of one triangle
- “twist“a subgraph / component
- When components are nested, try to maintain a constant spacing
- Bend placement: either next to a vertex or in the center of a set of vertices
- Drawing diagonally is not a bad idea? (Binary tree)
- ???