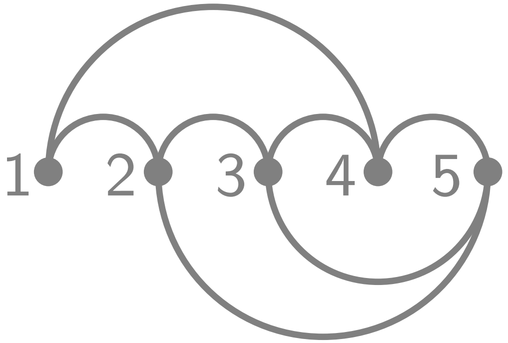
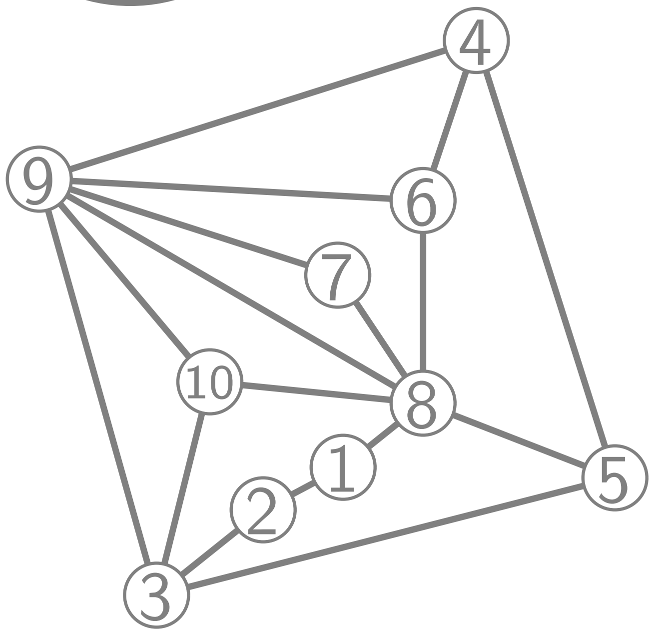
Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Visualization of Graphs

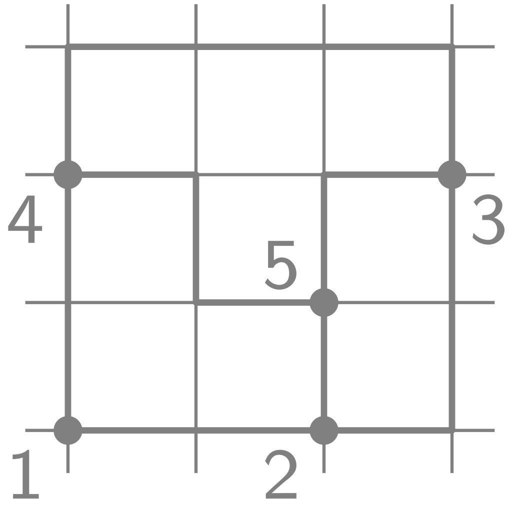
Lecture 1a:
The Graph Visualization Problem



Part I:
Organizational & Overview



Jonathan Klawitter



Organizational

Lectures:

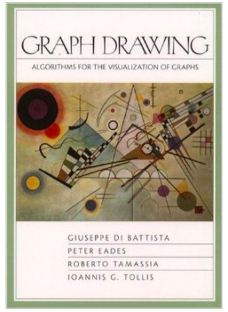
- Pre-recorded videos (as you see here)
- Release date: Weekend before
- Thursday 10:15 – 11:15: Questions/Discussion in Zoom
- Questions/Tasks in the Videos

Tutorials:

- One sheet per lecture
- 20 Points per sheet
- Scoring 50% overall $\Rightarrow$ bonus
- Submit solutions online
- Recommend LaTeX (template provided)
- Discussion and solutions ..

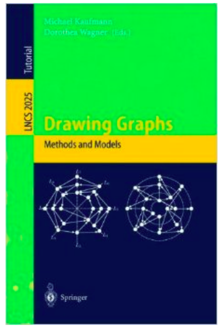
Books

[GD]



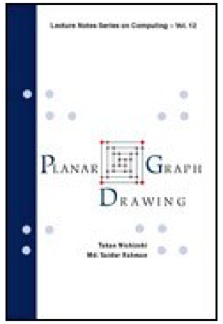
G. Di Battista, P. Eades, R. Tamassia, I. Tollis:
Graph Drawing: Algorithms for the Visualization of Graphs
Prentice Hall, 1998

[DG]



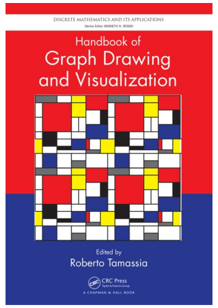
M. Kaufmann, D. Wagner:
Drawing Graphs: Methods and Models
Springer, 2001

[PGD]



T. Nishizeki, Md. S. Rahman:
Planar Graph Drawing
World Scientific, 2004

[HGDV]



R. Tamassia:
Handbook of Graph Drawing and Visualization
CRC Press, 2013
`http://cs.brown.edu/people/rtamassi/gdhandbook/`

What is this course about?

Learning objectives

- Overview of graph visualization
- Improved knowledge of modeling and solving problems via graph algorithms

Visualization problem:

- Given a graph G, visualize it with a drawing Γ

Here:

- Reducing the visualisation problem to its **algorithmic core**

graph class $\Rightarrow$ layout style $\Rightarrow$ algorithm $\Rightarrow$ analysis

- modeling
- data structures
- divide & conquer, incremental
- combinatorial optimization (flows, ILPs)
- force-based algorithm
- proofs

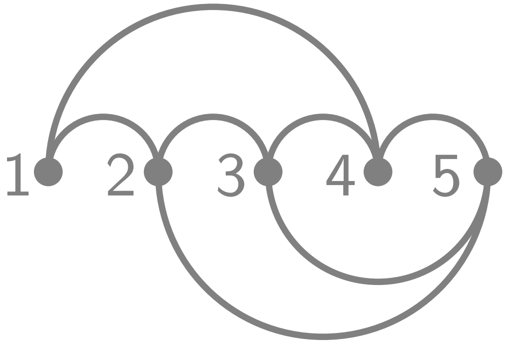
What is this course about?

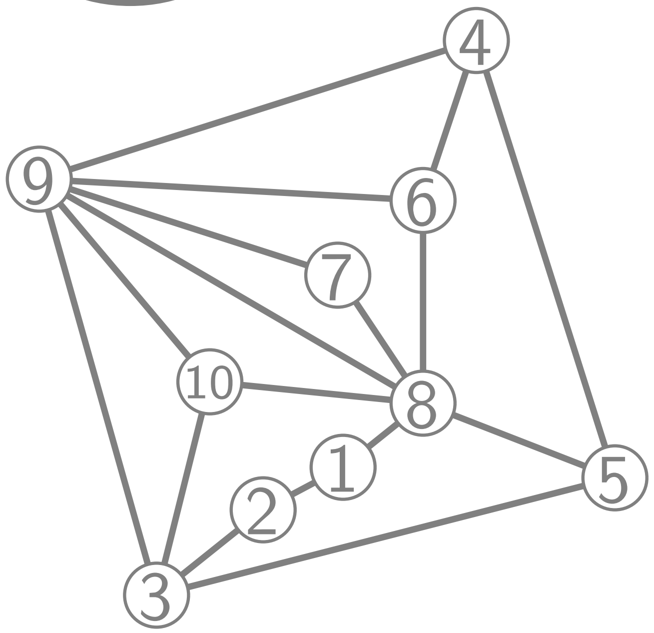
Topics

- Drawing Trees and Series-Parallel Graphs
- Tutte Embedding and Force-Based Drawing Algorithms
- Straight-Line Drawings of Planar Graphs
- Orthogonal Grid Drawings
- Octilinear Drawings for Metro Maps
- Upwards Planar Drawings
- Hierarchical Layouts of Directed Graphs
- Contact Representations
- Visibility Representations
- The Crossing Lemma
- Beyond Planarity

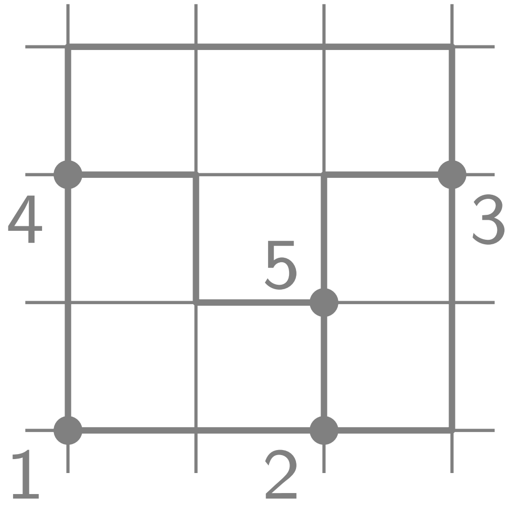
Visualization of Graphs

Lecture 1a:
The Graph Visualization Problem





Part II:
The Layout Problem



Jonathan Klawitter

Graphs and their representations

What is a graph?

- graph $G = (V, E)$
- vertices $V = \{v_1, v_2, \ldots, v_n\}$
- edge $E = \{e_1, e_2, \ldots, e_m\}$

Representation?

- Set notation

$$
\begin{aligned}
V &= \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\} \\
E &= \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \\
&\quad \{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \\
&\quad \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \\
&\quad \{v_9, v_{10}\}\}
\end{aligned}
$$

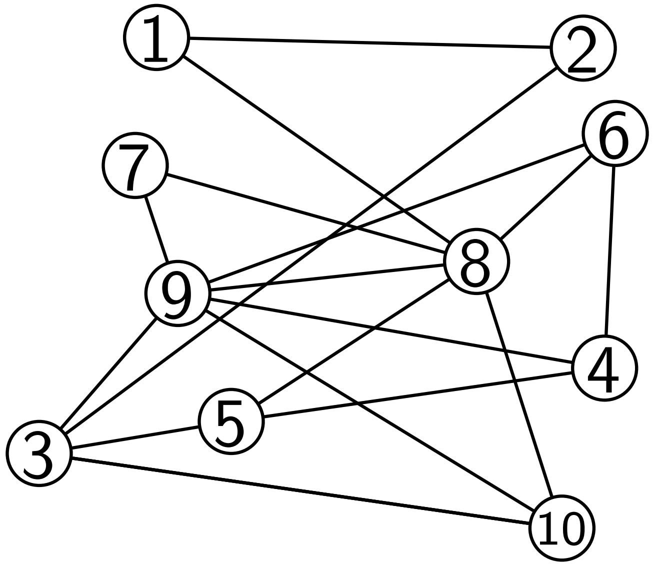
- Adjacency list

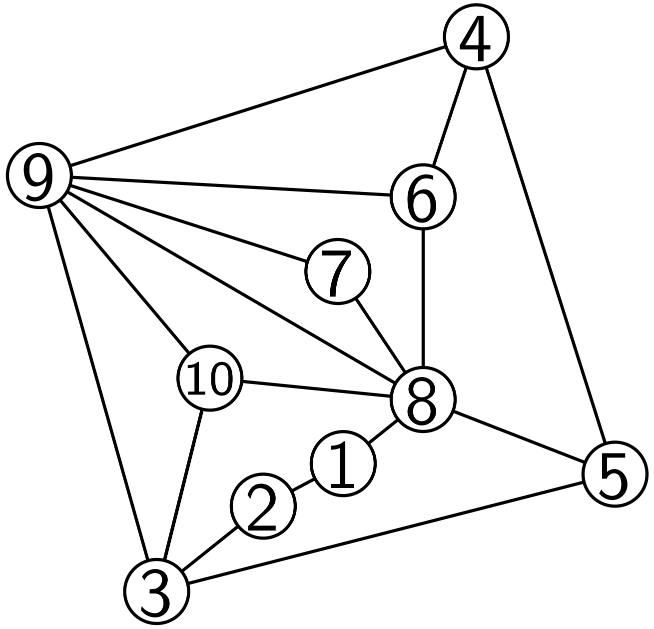
v_1:	v_2, v_8	v_6:	v_4, v_8, v_9
v_2:	v_1, v_3	v_7:	v_8, v_9
v_3:	v_2, v_5, v_9, v_{10}	v_8:	$v_1, v_5, v_6, v_7, v_9, v_{10}$
v_4:	v_5, v_6, v_9	v_9:	$v_3, v_4, v_6, v_7, v_8, v_{10}$
v_5:	v_3, v_4, v_8	v_{10}:	v_3, v_8, v_9

- Adjacency matrix

$$
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{pmatrix}
$$

- Drawing





Why draw graphs?

Graphs are a mathematical representation of real physical and abstract networks.

Abstract networks

- Social networks
- Communication networks
- Phylogenetic networks
- Metabolic networks
- Class/Object Relation Digraphs (UML)
- ...

Physical networks

- Metro systems
- Road networks
- Power grids
- Telecommunication networks
- Integrated circuits
- ...

Why draw graphs?

Graphs are a mathematical representation of real physical and abstract networks.

- **People think visually** – complex graphs are hard to grasp without good visualisations!

Why draw graphs?

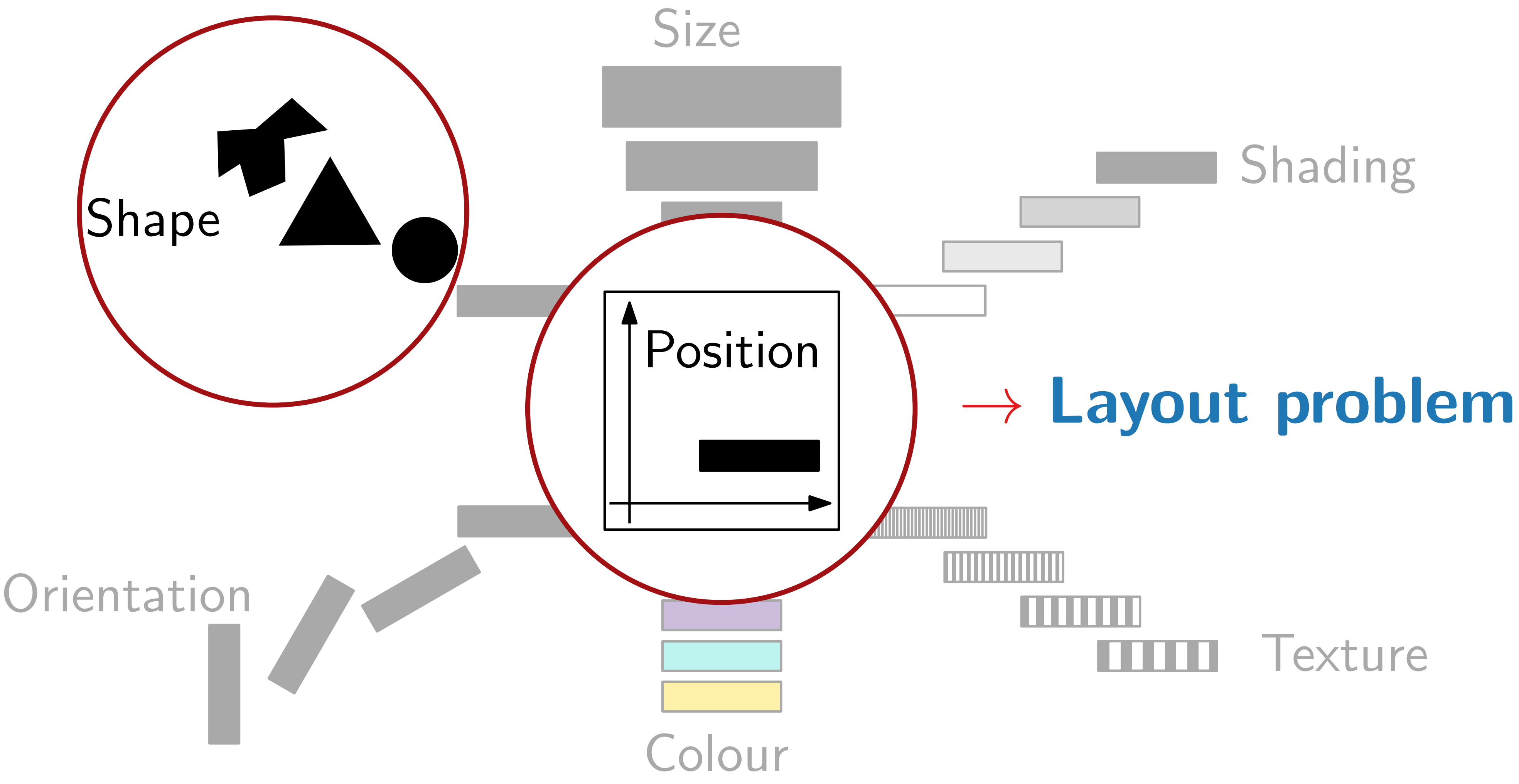
Graphs are a mathematical representation of real physical and abstract networks.

- **People think visually** – complex graphs are hard to grasp without good visualisations!
- Visualisations help with the **communication** and **exploration** of networks.
- Some graphs are too big to draw them by hand.

We need algorithms that draw graphs automatically to make networks more accessible to humans.

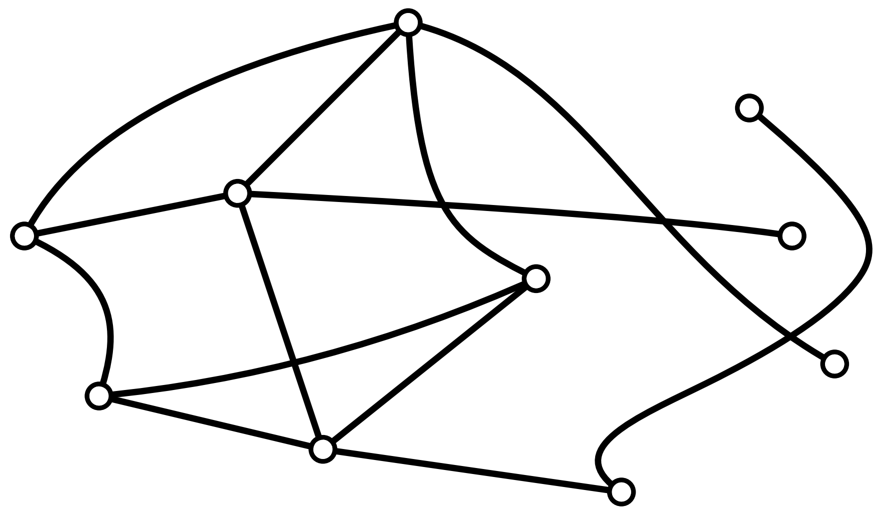
What are we interested in?

- Jacques Bertin defined visualising variables (1967)



The layout problem?

- *Here* restricted to the **standard representation**, so-called node-link diagrams.



Graph Visualization Problem

in: Graph $G = (V, E)$

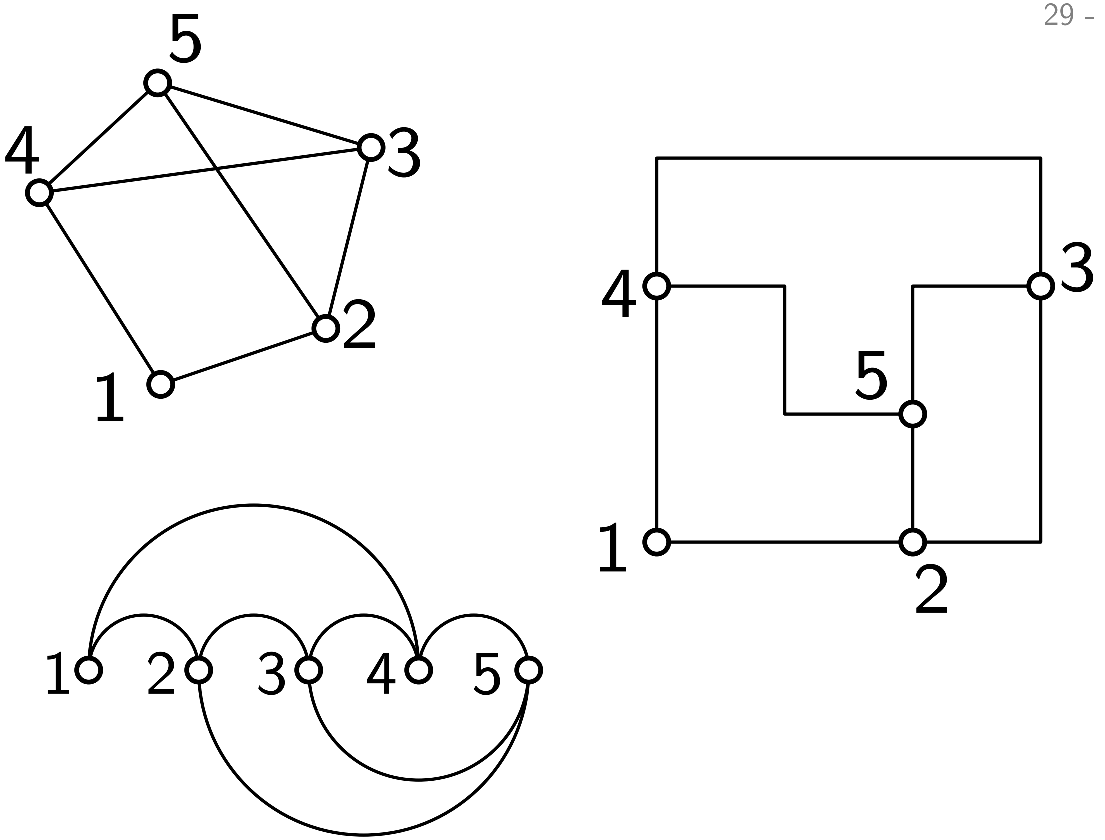
out: **nice** drawing Γ of G

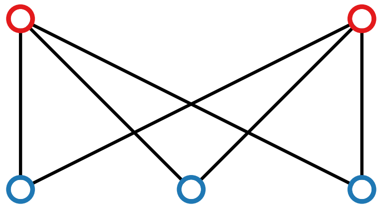
- $\Gamma \colon V \to \mathbb{R}^2$, vertex $v \mapsto$ point $\Gamma(v)$
- $\Gamma \colon E \to$ curves in $\mathbb{R}^2$, edge $\{u, v\} \mapsto$ simple, open curve $\Gamma(\{u, v\})$ with endpoints $\Gamma(u)$ und $\Gamma(v)$

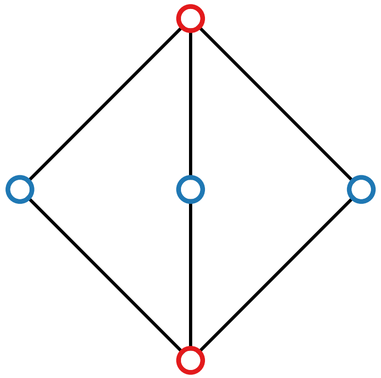
But what is a **nice** drawing?

Requirements of a graph layout

1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimized, e.g.
   - crossing/bend minimization

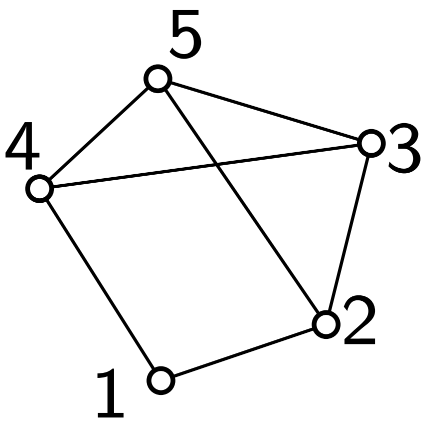


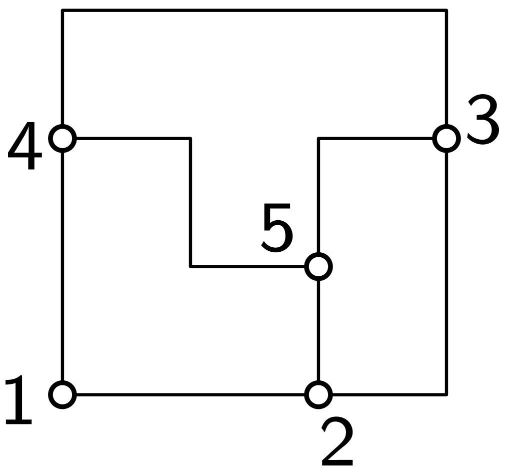


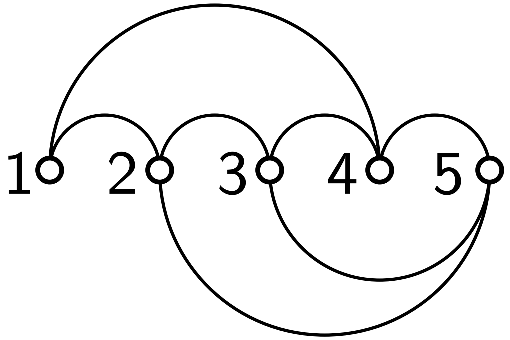


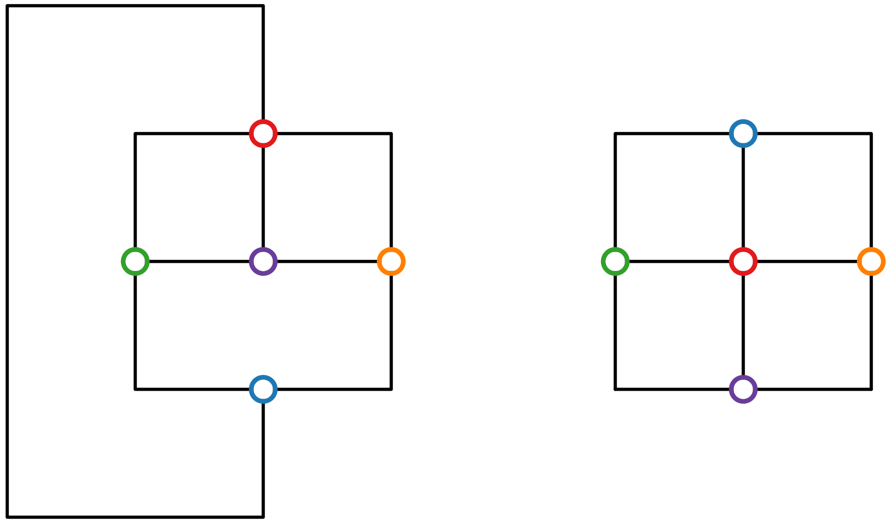
Requirements of a graph layout

1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimized, e.g.
   - crossing/bend minimization



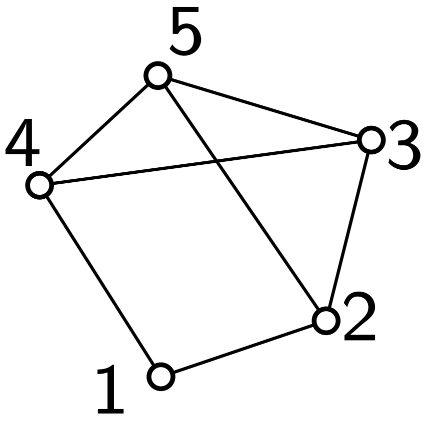


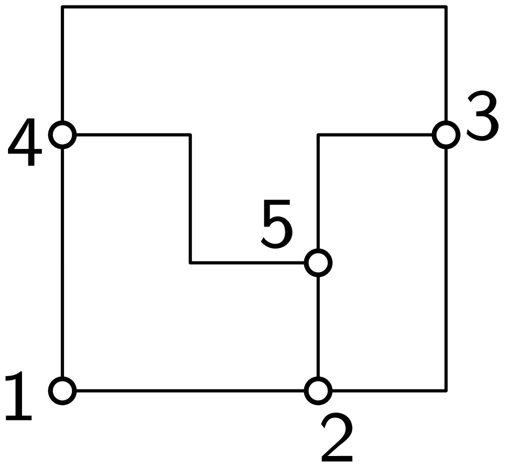


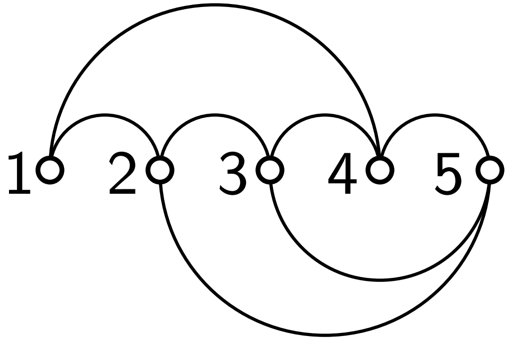


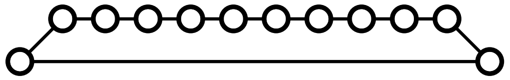
Requirements of a graph layout

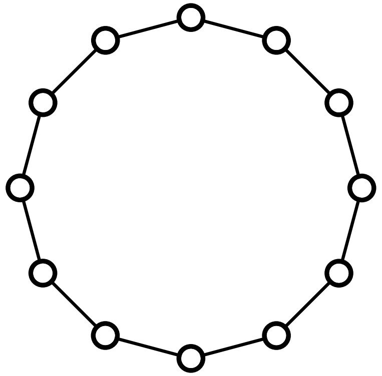
1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimized, e.g.
   - crossing/bend minimization
   - edge length uniformity





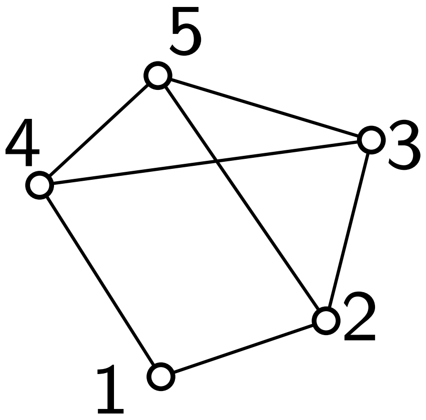


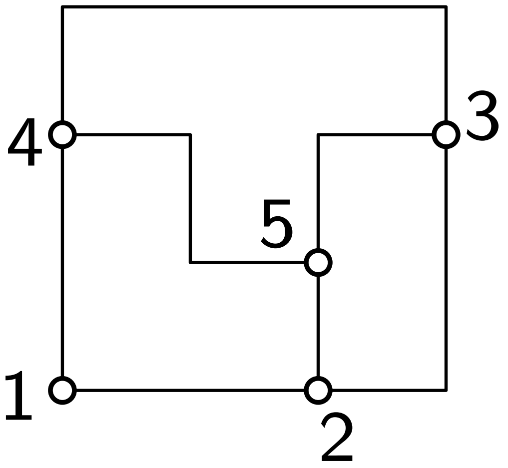


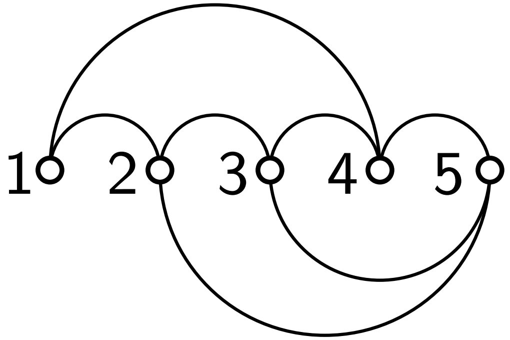


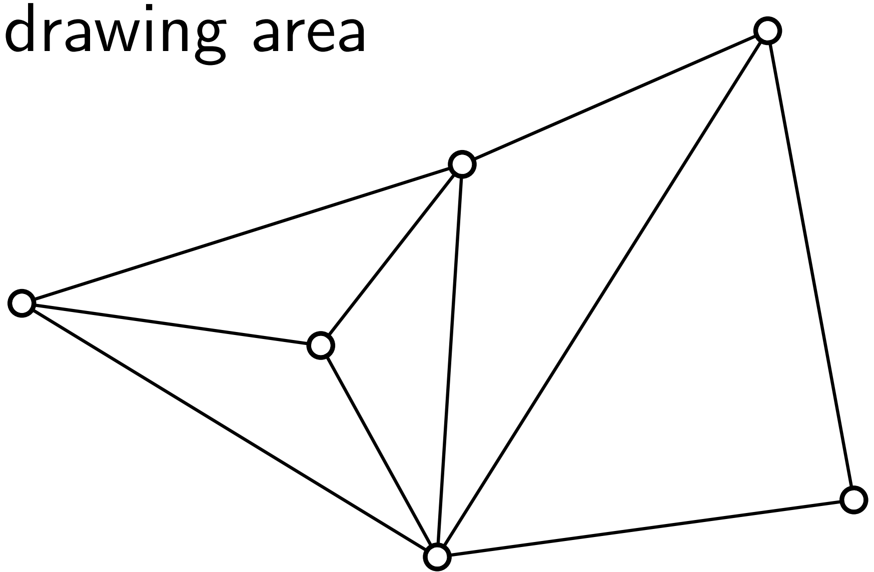
Requirements of a graph layout

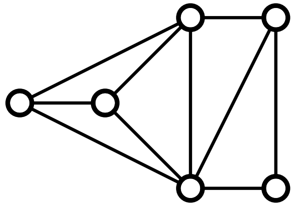
1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimized, e.g.
   - crossing/bend minimization
   - edge length uniformity
   - minimizing total edge length/drawing area





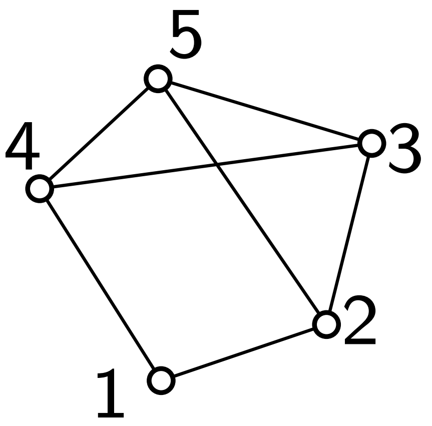


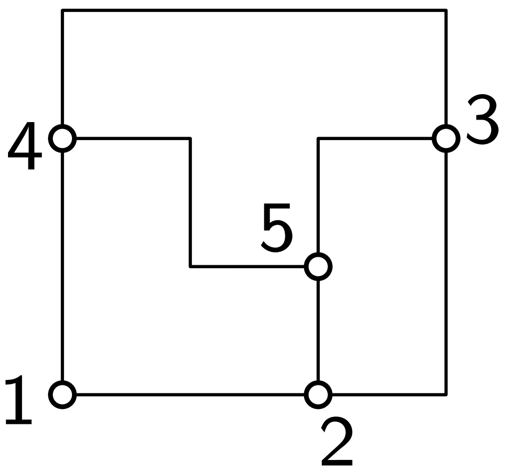


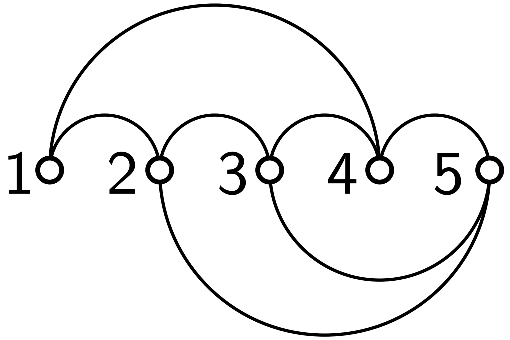


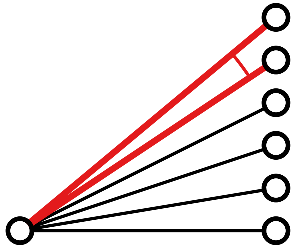
Requirements of a graph layout

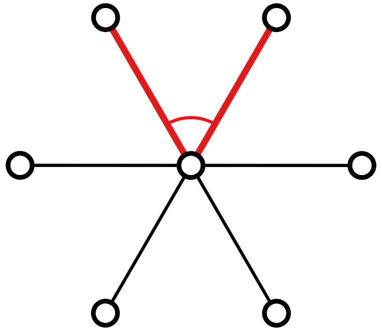
1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimized, e.g.
   - crossing/bend minimization
   - edge length uniformity
   - minimizing total edge length/drawing area
   - angular resolution





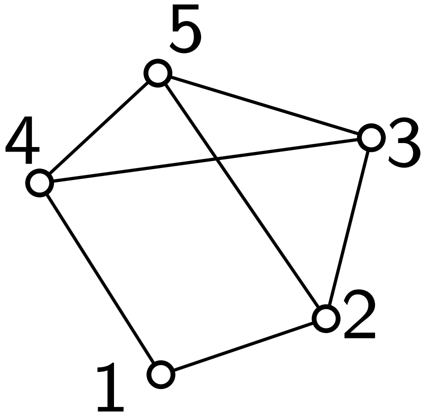


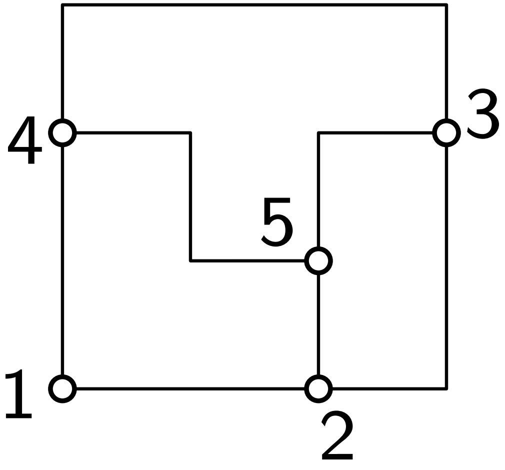


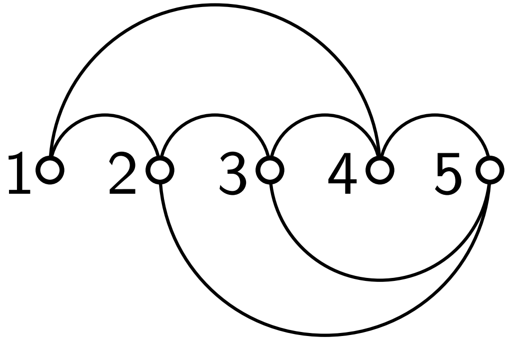


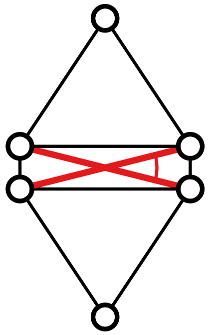
Requirements of a graph layout

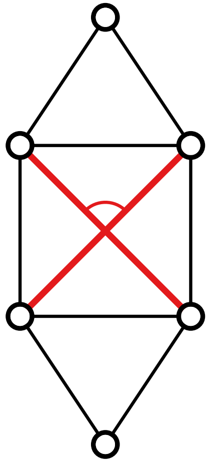
1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimized, e.g.
   - crossing/bend minimization
   - edge length uniformity
   - minimizing total edge length/drawing area
   - angular resolution





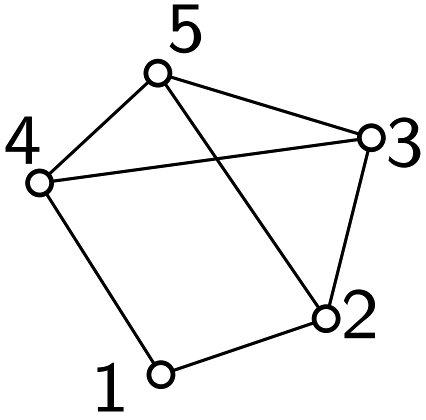


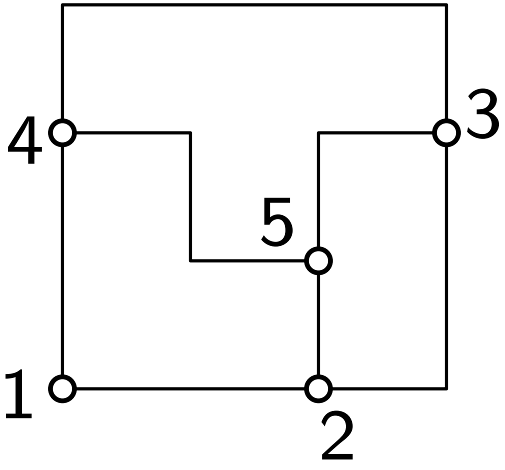


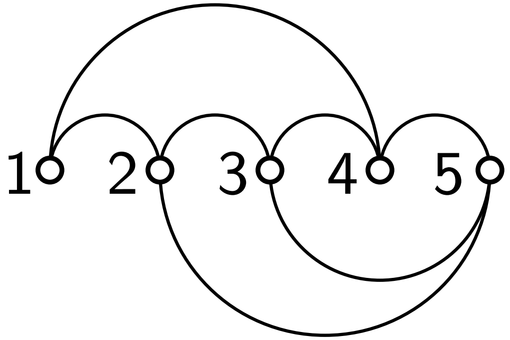


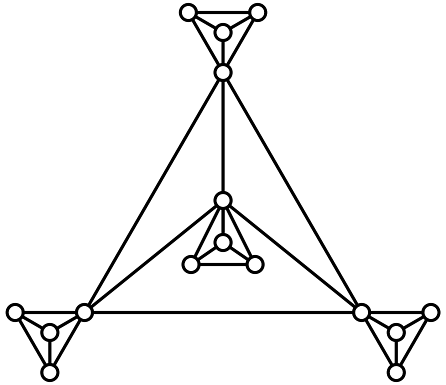
Requirements of a graph layout

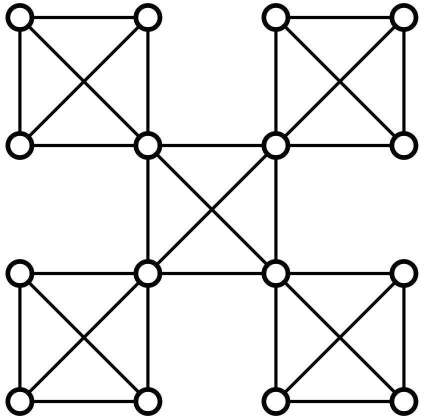
1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimized, e.g.
   - crossing/bend minimization
   - edge length uniformity
   - minimizing total edge length/drawing area
   - angular resolution
   - symmetry/structure





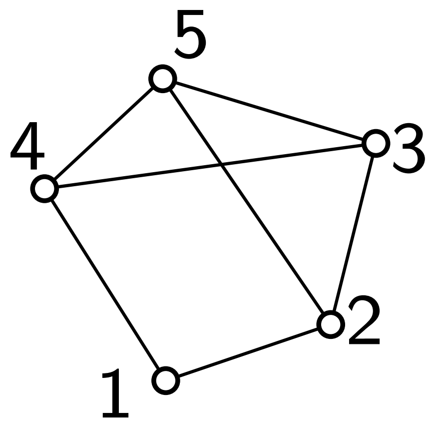


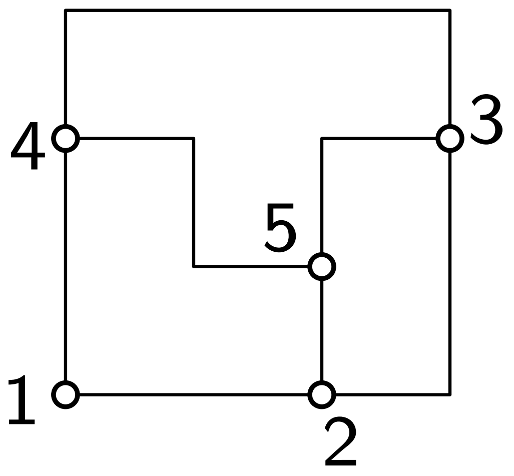


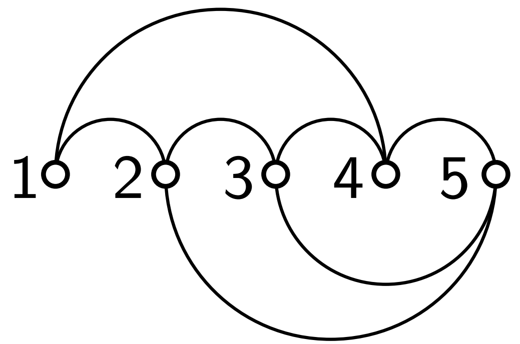


Requirements of a graph layout

1. Drawing conventions and requirements, e.g.,
   - straight edges with $\Gamma(uv) = \overline{\Gamma(u)\Gamma(v)}$
   - orthogonal edges (i.e. with bends)
   - grid drawings
   - without crossing
2. Aesthetics to be optimized, e.g.
   - crossing/bend minimization
   - edge length uniformity
   - minimizing total edge length/drawing area
   - angular resolution
   - symmetry/structure
3. Local Constraints, e.g.
   - restrictions on neighboring vertices (e.g., "upward").
   - restrictions on groups of vertices/edges (e.g., "clustered").







→ such criteria are often inversely related

→ lead to NP-hard optimization problems

The layout problem

Graph visualisation problem

in: Graph $G = (V, E)$

out: Drawing Γ of G such that

- **drawing conventions** are met,
- **aesthetic criteria** are optimised, and
- some **additional constraints** are satisfied.