Algorithms for Graph Visualization

Introduction

Tamara Mchedlidze, Marcel Radermacher
16.10.2018
Introduction

Lectures

- Tamara Mchedlidze
- mched@iti.uka.de
- Office 307
- Office hours: request by email

Exercises

- Marcel Radermacher
- radermacher@kit.edu
- Office 306
- Office hours: request by email
Introduction

Lectures

• Tamara Mchedlidze
• mched@iti.uka.de
• Office 307
• Office hours: request by email

Exercises

• Marcel Radermacher
• radermacher@kit.edu
• Office 306
• Office hours: request by email

YOU: Name, Field of your Bachelor studies, why you are interested in this lecture
Introduction

Lectures

- Tamara Mchedlidze
- mched@iti.uka.de
- Office 307
- Office hours: request by email

Exercises

- Marcel Radermacher
- radermacher@kit.edu
- Office 306
- Office hours: request by email

YOU: Name, Field of your Bachelor studies, why you are interested in this lecture

Mailing list
Overview

About this course

Repetition of the material. We build our Mind Map.

Drawing graphs “by hand”. Complete MindMap.

Formal definition of Layout Problem.

About this course: learning objectives.

Applications gallery.
Overview

About this course

Repetition of the material. We build our Mind Map.

Drawing graphs “by hand”. Complete MindMap.

Formal definition of Layout Problem.

About this course: learning objectives.

Applications gallery.
About this Course

• **Lecture:** Wednesday 14:00 – 15:30, Room 301
• **Exercise:** Tuesday 14:00 – 15:30, Room 301
• exact plan on the web-page

**Webseite**

i11www.itl.kit.edu/teaching/winter2018/graphvis/

• Latest news
• Lecture slides
• Exercise sheets
• Literature & Additional material
• Lecture notes (skript)
About this Course

Media:
• **Slides** & Blackboard & Pinboard
• Exercise sheets are provided (at least) a week before the exercise session
• (incomplete) Lecture notes/Books
• Original literature (papers)
Books (available in the library)

G. di Battista, P. Eades, R. Tamassia, I. Tollis: Graph Drawing
Prentice Hall, 1998

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

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

R. Tamassia: Handbook of Graph Drawing and Visualization
CRC Press, 2013

http://cs.brown.edu/~rt/gdhandbook/
About this Course

Master Informatics

- Module: General: M-INFO-102094
  This year: T-INFO-104390
About this Course

Master Informatics

• Module: General: M-INFO-102094
  This year: T-INFO-104390

Suggested time requirements:

• Attending Lecture and Exercises: ca. 35h
• Preparation/post-processing ca. 35h
• Work on the exercises ca. 40h
• Preparation for the exam ca. 40h

5LP = 150h
About this Course

Master Informatics
• Module: General: M-INF-102094
  This year: T-INF-104390

Suggested time requirements: 5LP = 150h
• Attending Lecture and Exercises: ca. 35h
• Preparation/post-processing ca. 35h
• Work on the exercises ca. 40h
• Preparation for the exam ca. 40h

Exercises:
• We expect that you participate actively in the exercise sessions (e.g. present your own solutions on the board)
• Submit a visualization for the practical task (bonus)

Examination procedure: Oral exam(app. 20 Minutes)
Overview

About this course

Repetition of the material. We build our Mind Map.

Drawing graphs “by hand”. Complete MindMap.

Formal definition of Layout Problem.

About this course: learning objectives.

Applications gallery.
Graph and its Representation

What is a Graph?
Graph and its Representation

What is a Graph?

Tuple $G = (V, E)$
Set of nodes $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\}, \ 1 \leq i \leq m, \ 1 \leq j, k \leq n$
Graph and its Representation

What is a Graph?

Tuple $G = (V, E)$
Set of nodes $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\}$, $1 \leq i \leq m$, $1 \leq j, k \leq n$

Representations?
Graph and its Representation

What is a Graph?

Tuple $G = (V, E)$
Set of nodes $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\}, \ 1 \leq i \leq m, \ 1 \leq j, k \leq n$

Representations?

Set representation:

$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$
$E = \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\},$
$\{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\},$
$\{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\},$
$\{v_9, v_{10}\}\}$
Graph and its Representation

What is a Graph?

Tuple $G = (V, E)$
Set of nodes $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\}$, $1 \leq i \leq m$, $1 \leq j, k \leq n$

Representations?

Set representation

Adjacency list

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

What is a Graph?

Tuple $G = (V, E)$
Set of nodes $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\}$, $1 \leq i \leq m$, $1 \leq j, k \leq n$

Representations?

Set representation
Adjacency list
Adjacency matrix

\[
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0
\end{pmatrix}
\]
Graph and its Representation

What is a Graph?

Tuple $G = (V, E)$
Set of nodes $V = \{v_1, \ldots, v_n\}$
Set of edges $E = \{e_1, \ldots, e_m\}$,
$e_i = \{v_j, v_k\}, 1 \leq i \leq m, 1 \leq j, k \leq n$

Representations?

Set representation
Adjacency list
Adjacency matrix
**Drawing** or Node-link diagram
Graph and its Representation

What is a Graph?

Tuple \( G = (V, E) \)

Set of nodes \( V = \{v_1, \ldots, v_n\} \)

Set of edges \( E = \{e_1, \ldots, e_m\} \),

\[ e_i = \{v_j, v_k\}, \quad 1 \leq i \leq m, \quad 1 \leq j, k \leq n \]

Representations?

Set representation

Adjacency list

Adjacency matrix

**Drawing** or Node-link diagram
Graph and its Representation

\[ V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\} \]
\[ E = \{\{v_1, v_2\}, \{v_1, v_8\}, \{v_2, v_3\}, \{v_3, v_5\}, \{v_3, v_9\}, \{v_3, v_{10}\}, \{v_4, v_5\}, \{v_4, v_6\}, \{v_4, v_9\}, \{v_5, v_8\}, \{v_6, v_8\}, \{v_6, v_9\}, \{v_7, v_8\}, \{v_7, v_9\}, \{v_8, v_{10}\}, \{v_9, v_{10}\}\} \]

\[
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0
\end{bmatrix}
\]

\[
v_1 : v_2, v_8 \\
v_2 : v_1, v_3 \\
v_3 : v_2, v_5, v_9, v_{10} \\
v_4 : v_5, v_6, v_9 \\
v_5 : v_3, v_4, v_8 \\
v_6 : v_4, v_8, v_9 \\
v_7 : v_8, v_9 \\
v_8 : v_1, v_5, v_6, v_7, v_9, v_{10} \\
v_9 : v_3, v_4, v_6, v_7, v_8, v_{10} \\
v_{10} : v_3, v_8, v_9
\]
Think and write down
• Why do we need node-link diagrams?
Why to draw graphs?

- Use human cognition efficiently.
- To be aware of network structure.
- To understand/reveal the structure.
- Explore the space to depict information.
- Interpret information about the graph.
- Communicate information.
- Classify graphs to classes.
Let’s Recall

Discuss with your neighbour or in groups of three and then write down

Graph classes you know (planar etc.)

Algorithmic techniques you know (greedy etc.)

Applications of network visualization you have heard about

We will group your knowledge into a MIND MAP
Let’s Recall

Discuss with your neighbour or in groups of three and then write down

Graph classes you know (planar etc.)

Algorithmic techniques you know (greedy etc.)

Applications of network visualization you have heard about

We will group your knowledge into a MIND MAP

Prerequisites: Algorithms 1 & 2, Theoretical Basics of inf.
Helpful: Algorithms for Planar Graphs
Overview

About this course

Repetition of the material. We build our Mind Map.

**Drawing graphs “by hand”. Complete MindMap.**

Formal definition of Layout Problem.

About this course: learning objectives.

Applications gallery.
How to draw graphs?

Work with your neighbour or in groups of three

- graphs in form of adjacency matrix/list
- Use https://www.yworks.com/downloads#yEd or paper
- draw all or some graphs as nice and as readable as possible
- export to PNG or make a picture and send to mched@iti.uka.de

We will show and discuss the results afterwards and complete the MIND MAP
Overview

About this course

Repetition of the material. We build our Mind Map.

Drawing graphs “by hand”. Complete MindMap.

Formal definition of Layout Problem.

About this course: learning objectives.

Applications gallery.
Visual Variables according to Bertin (1967)

- **Shape**
- **Orientation**
- **Position**
- **Size**
- **Value**
- **Texture**
- **Color**
Visual Variables according to Bertin (1967)

- position
- value
- texture
- size
- shape
- orientation
- color

→ Layout problem
Visual Variables according to Bertin (1967)

- shape
- orientation
- size
- position
- color
- texture
- value

→ Layout problem
Layout Problem

Graph visualization problem

given : Graph $G = (V, E)$
find: good drawing $\Gamma$ of $G$

• $\Gamma : V \rightarrow \mathbb{R}^2$, nodes $v \mapsto$ point $\Gamma(v)$
• $\Gamma : E \rightarrow$ curves in $\mathbb{R}^2$, edge $\{u, v\} \mapsto$ simple open curve $c_{uv} : [0, 1] \rightarrow \mathbb{R}^2$ where $c_{uv}(0) = \Gamma(u)$ and $c_{uv}(1) = \Gamma(v)$
Layout Problem

Graph visualization problem

given: Graph $G = (V, E)$
find: good drawing $\Gamma$ of $G$

• $\Gamma : V \rightarrow \mathbb{R}^2$, nodes $v \mapsto$ point $\Gamma(v)$
• $\Gamma : E \rightarrow$ curves in $\mathbb{R}^2$, edge $\{u, v\} \mapsto$ simple open curve $c_{uv} : [0, 1] \rightarrow \mathbb{R}^2$ where $c_{uv}(0) = \Gamma(u)$ and $c_{uv}(1) = \Gamma(v)$
1) **Drawing conventions**, required properties, for example

- straight-line edges
- orthogonal edges (with bends 90 degrees)
- Drawing on a grid
- crossing-free
- ....
Layout Problem

1) **Drawing conventions**, required properties

2) **Aesthetics** (to be optimized), for example:
   - Number of crossing
   - Number of bends
   - Uniform edge length
   - Area/length
   - Angular resolution
   - Symmetry
   - ...
Layout Problem

1) **Drawing conventions**, required properties

2) **Aesthetics** (to be optimized)

3) **Partial/local constraints**, for example:
   - Positions of several vertices
   - Relative positions of vertices
   - Group of nodes drawn close to each other
Layout Problem – Second Attempt

Graph visualization problem

given: Graph $G = (V, E)$
find: a drawing $\Gamma$ of $G$, that
  • complies with drawing conventions
  • optimizes aesthetics
  • satisfies local/partial constraints
Layout Problem – Second Attempt

Graph visualization problem

given: Graph $G = (V, E)$
find: a drawing $\Gamma$ of $G$, that
• complies with drawing conventions
• optimizes aesthetics
• satisfies local/partial constraints

→ often lead to NP-hard optimization problems!
→ often several competing criteria
Overview

About this course

Repetition of the material. We build our Mind Map.

Drawing graphs “by hand”. Complete MindMap.

Formal definition of Layout Problem.

About this course: learning objectives.

Applications gallery.
Learning Objectives

At the end of the semester you are able to:
Learning Objectives

At the end of the semester you are able to:

• List various network visualization styles
• Formally state a network visualization problem
• Describe several algorithms for network visualization in an intuitive way
• Describe formally several network visualization algorithms
• Identify the techniques behind the algorithms (greedy, iterative, dynamic programming, etc.)
• Analyze the time complexity of algorithms
• Proof correctness of the algorithms
• Use a tool or library to produce a network visualization
• Solve new network visualization problems by selecting and adapting known approaches
Learning Objectives

At the end of the semester you are able to:

- List various network visualization styles
- Formally state a network visualization problem
- Describe several algorithms for network visualization in an intuitive way
- Describe formally several network visualization algorithms
- Identify the techniques behind the algorithms (greedy, iterative, dynamic programming, etc.)
- Analyze the time complexity of algorithms
- Proof correctness of the algorithms
- Use a tool or library to produce a network visualization
- Solve new network visualization problems by selecting and adapting known approaches
## Learning Objectives

At the end of the semester you are able to:

<table>
<thead>
<tr>
<th>Recall Level</th>
<th>Analyze, Apply, Generalize Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>• List various network visualization styles</td>
<td>• Analyze, Apply, Generalize Level</td>
</tr>
<tr>
<td>• Formally state a network visualization problem</td>
<td>• Analyze, Apply, Generalize Level</td>
</tr>
<tr>
<td>• Describe several algorithms for network visualization in a intuitive way</td>
<td>• Solve new network visualization problems by selecting and adapting known approaches</td>
</tr>
<tr>
<td>• Describe formally several network visualization algorithms</td>
<td>• Analyze, Apply, Generalize Level</td>
</tr>
<tr>
<td>• Identify the techniques behind the algorithms (greedy, iterative, dynamic programming, etc.)</td>
<td>• Analyze, Apply, Generalize Level</td>
</tr>
<tr>
<td>• Analyze the time complexity of algorithms</td>
<td>• Analyze, Apply, Generalize Level</td>
</tr>
<tr>
<td>• Proof correctness of the algorithms</td>
<td>• Analyze, Apply, Generalize Level</td>
</tr>
<tr>
<td>• Use a tool or library to produce a network visualization</td>
<td>• Analyze, Apply, Generalize Level</td>
</tr>
<tr>
<td>• Solve new network visualization problems by selecting and adapting known approaches</td>
<td>• Analyze, Apply, Generalize Level</td>
</tr>
</tbody>
</table>
Overview

About this course

Repetition of the material. We build our Mind Map.

Drawing graphs “by hand”. Complete MindMap.

Formal definition of Layout Problem.

About this course: learning objectives.

Applications gallery.
Biblical characters and events (1202)

Source: Joachim de Fiore
”Tree of Life“ (1516)

Source: Paul Riccius, Portae Lucis
Geometrical Concepts (1587)

Source: Christophe de Savigny
Genealogical Tree (1879)

Source: Ernst Haeckel
Sociogramm (1933)

Source: Moreno, 1933
Social Network – Organization within UBS

* Member of the Group Executive Board
CPAN Developer-Graph

Source: cpan-explorer.org
last.fm Graph of musics as political map

(Gansner, Hu, Kobourov: GMap, 2009)
last.fm Graph of musics as political map

(Gansner, Hu, Kobourov: GMap, 2009)
Blogosphere 2004 Elections USA

Source: Adamic, Glance, 2005
Social Network – World Finance System

World Finance Corporation
© Mark Lombardi
Temporal Graph Layout: Storylines

Source: ABC news, Australia
Traffic network – Highways USA
Traffic network – Highways USA
Co-centric Tube Map

Source: Maxwell Roberts
Curvilinear S/U-bahn map

Source: KVV and Maxwell Roberts
Flight Connections
Flow-Map: Whiskey Export

Source: Verbeek, Buchin, Speckmann., 2011
Telephony Map

Source: TeleGeography
Monitoring of Energy Network

Source: Eir Grid, Ireland
Medicine – Diseases
Medicine – phylogenetic Tree

Kinesin-13 (MCAK/KIF2)
Kinesin-8 (Kip3)
Kinesin-7 (CENP-E)
At2
At1
Kinesin-3 (Unc104/KIF1)
Kinesin-4 (ChrKin/KIF4)
Kinesin-1 (KHC)
Kinesin-5 (BimC)
Kinesin-2 (KRP85/95)
Kinesin-6 (MKLP1)
Kinesin-14 (C-Terminal Motor)

50 changes

Elise M. Dagenbach & Sharyn A. Endow
Clustered Software-Graph in 3D

Source: Balzer, Deussen, 2007
Software Call-Graph with edge-bundling

Source: Danny Holten, 2011
Large Graphs – Object Mesh

Source: Yifan Hu
Alternative Visualizations: Explorer vs Treemap
Alternative Visualizations: Contact map
Tools

Libraries for graph visualization

• JUNG jung.sourceforge.net (Java)
• OGDF www.ogdf.net (C++)

Visualization tools

• visone visone.info
• graphviz www.graphviz.org
• yEd www.yworks.com
• Gephi www.gephi.org
Next Meeting

**Lectures** 17.10, 30.10 14:00
**Exercise** on 24.10 14:00
Next Meeting

**Lectures** 17.10, 30.10 14:00  
**Exercise** on 24.10 14:00

**Topic** Tree Layouts

**Home task** In which applications we need to construct a tree layout?