1 Datenstrukturen

1.1 Abstrakte Datentypen

1.2 Lineare Strukturen

1.3 Bäume

1.4 Prioritätsschlangen

1.5 Graphen

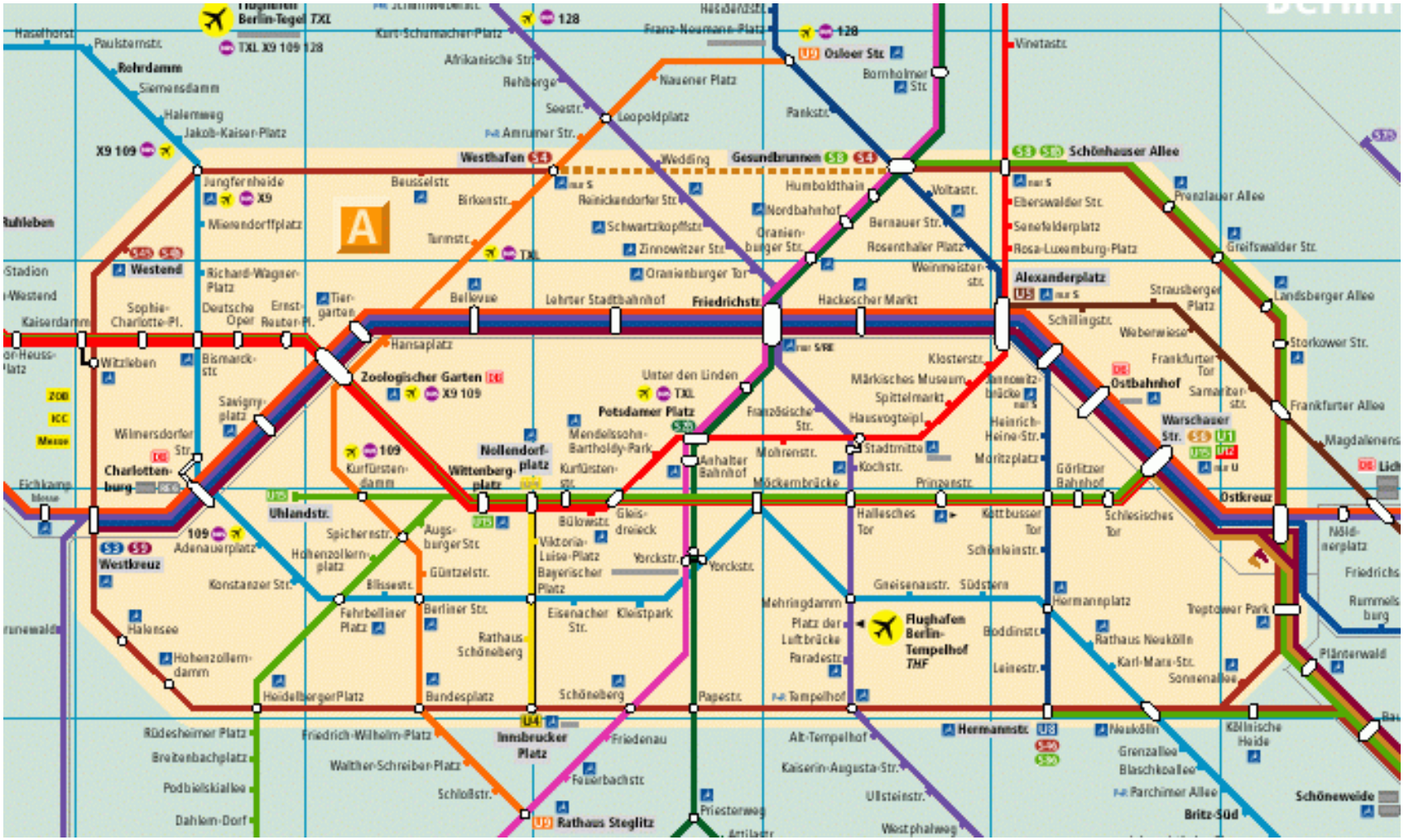
1.5 Graphen

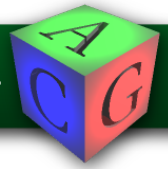
- Darstellung allgemeiner Beziehungen zwischen Objekten/Elementen
  - Objekte = Knoten:
    $V = \{v_1, \ldots, v_n\}$
  - Beziehungen = Kanten:
    $E = \{(a_1,b_1), \ldots (a_m,b_m)\}$
- Kanten können gerichtet (Vorgänger/ Nachfolger) oder ungerichtet (Nachbarschaft) definiert werden.

Beispiele

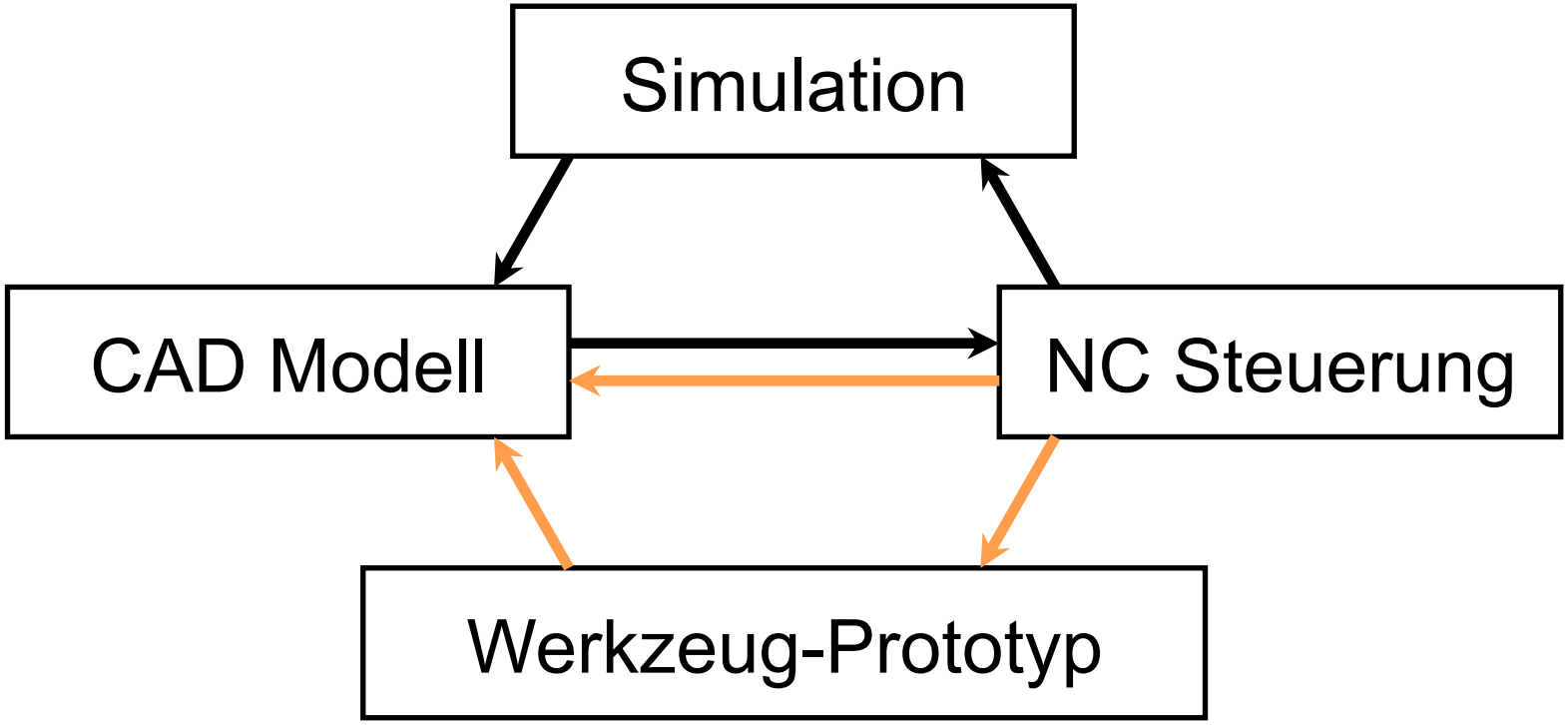
- Flugverbindungen (ungerichtet)
- Straßennetz in Aachen (gerichtet)
- Abhängigkeiten von Arbeitspaketen in einem Großprojekt (gerichtet)
- 3D Polygonmodelle in der Computergraphik (ungerichtet)
- Querbezüge in einem Hypertext-Dokument (HTML) (gerichtet)
- ...

Beispiele


Flughafen Berlin-Tegel TXL
Westhafen
Gesundbrunnen
Schönhauser Allee
Zoologischer Garten
Potsdamer Platz
Alexanderplatz
Ostbahnhof
Ostkreuz
Westkreuz
Charlottenburg
Uhlandstr.
Hallesches Tor
Flughafen Berlin-Tempelhof THF
Innsbrucker Platz
Rathaus Steglitz
Hermannstr.
Neukölln
Schöneweide
Britz-Süd

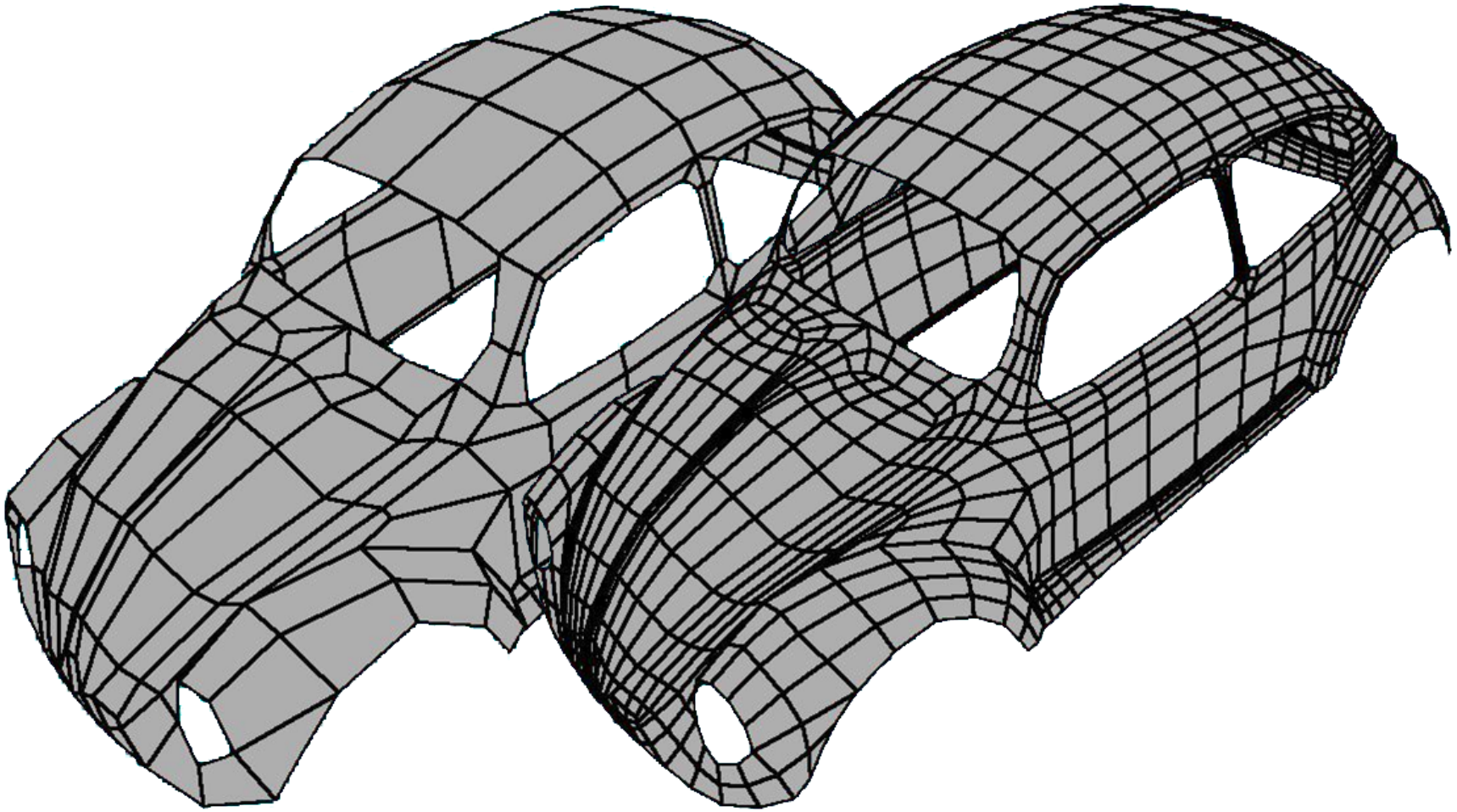


Beispiele



Workflow Mechatronic

Beispiele



Begriffe

- Zusammenhängend (stark)
- Vollständig
- Isomorph
- Planare Graphen

Begriffe

- Zusammenhängend (stark)
  - Für jedes Paar v_i, v_j von Knoten existiert ein Pfad von Kanten von v_i nach v_j oder (und) von v_j nach v_i.
- Vollständig
- Isomorph
- Planare Graphen

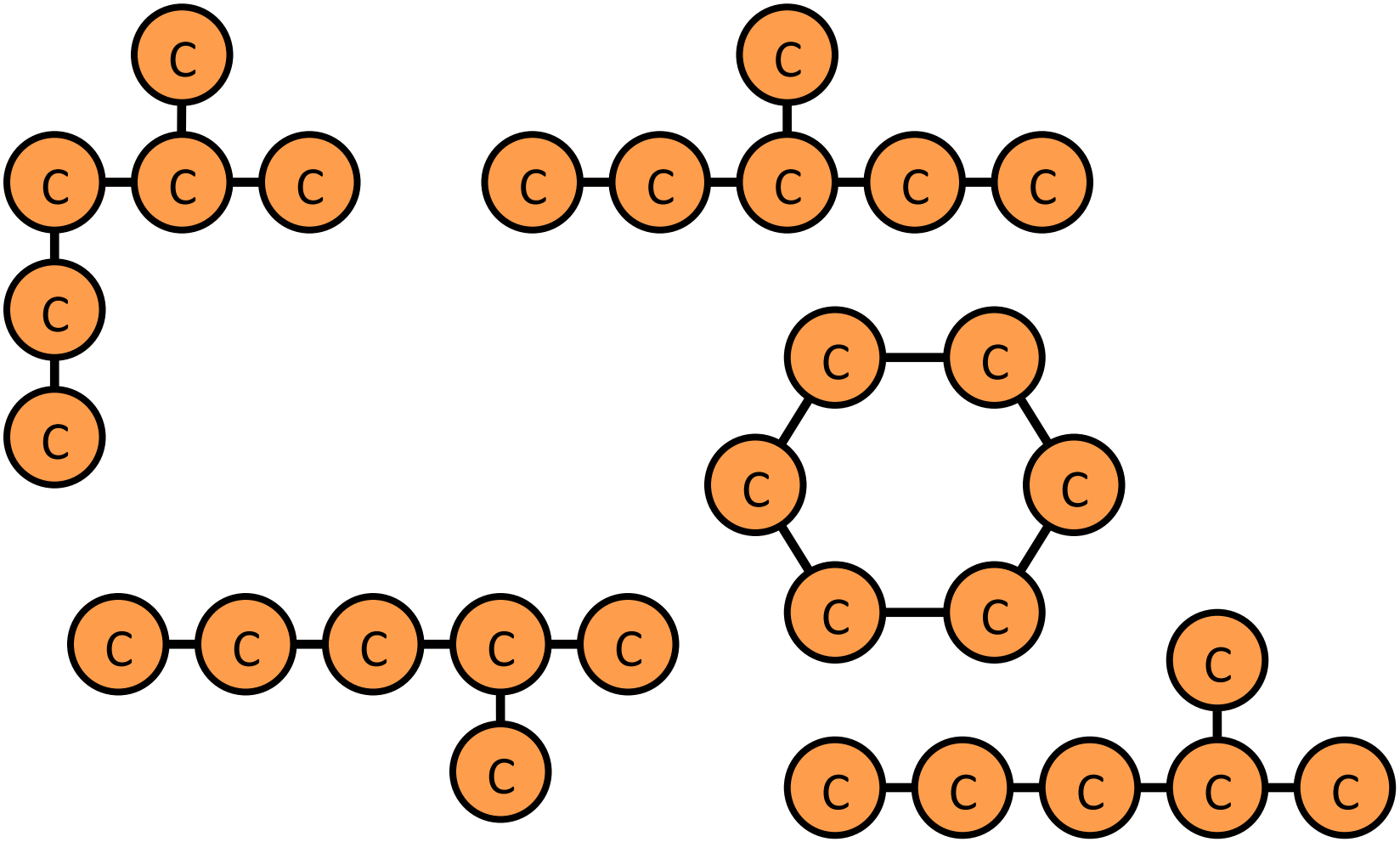
Begriffe

- Zusammenhängend (stark)
- Vollständig
  - Zwischen je zwei Knoten existiert eine Kante
  - $m = n \times (n-1)/2$
- Isomorph
- Planare Graphen

Begriffe

- Zusammenhängend (stark)
- Vollständig
- Isomorph
  - Graphen (V_1,E_1) und (V_2,E_2) durch Permutation der Knoten(indizes) ineinander überführbar.
  - Notwendige Kriterien: $n_1 = n_2$, $m_1 = m_2$, Anzahl der Knoten mit Grad k, ...
- Planare Graphen

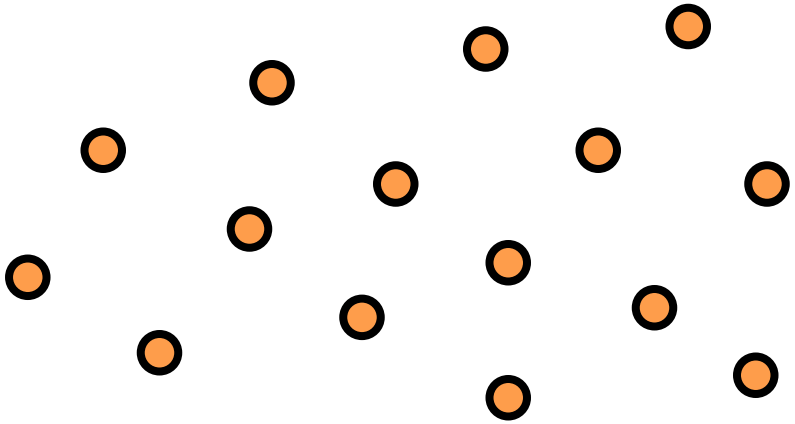
Kohlenwasserstoffe



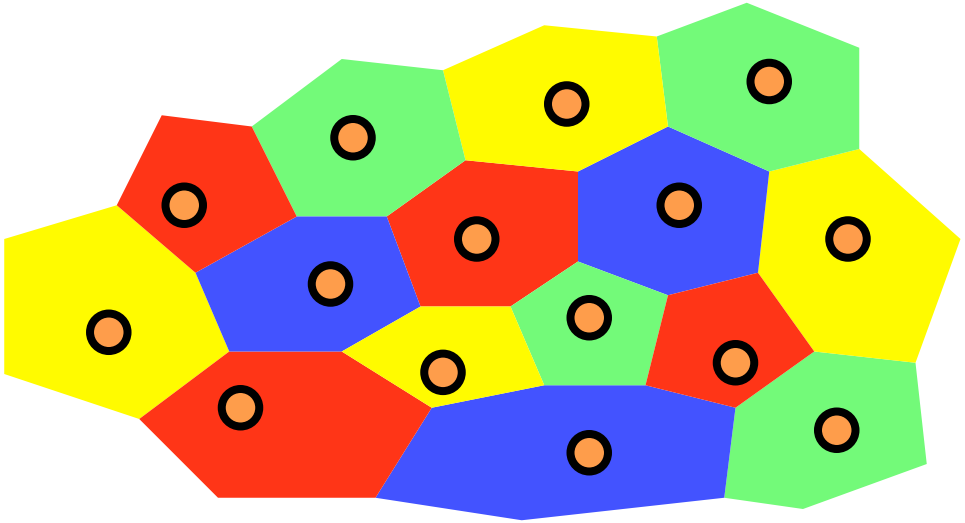
Begriffe

- Zusammenhängend (stark)
- Vollständig
- Isomorph
- Planare Graphen
  - Zerlegung der Ebene in disjunkte Zellen
    - Kompatibilitätsbedingungen zwischen Kanten

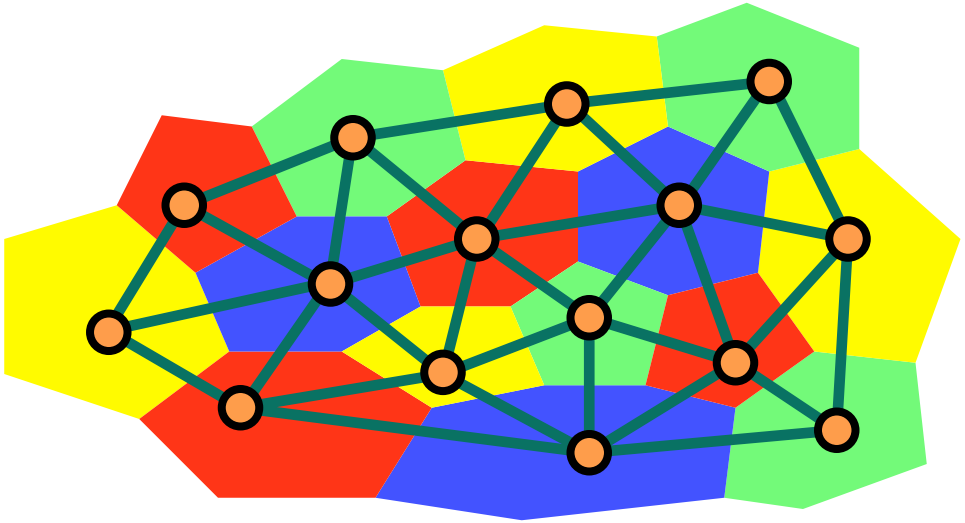
Mobilfunknetze



Mobilfunknetze



Mobilfunknetze



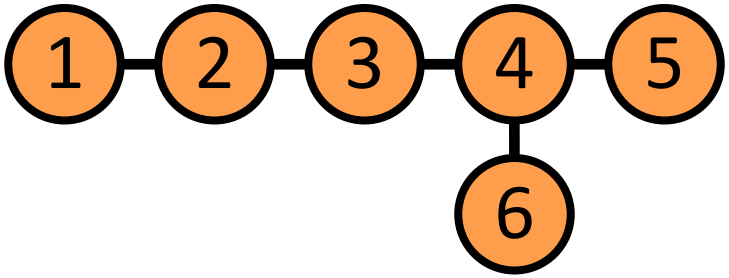
Repräsentationen

- Allgemeine Graphen
  - Adjazenzmatrix
  - Adjazenzliste
- Planare Graphen
- Simplex-Strukturen

Adjazenzmatrix

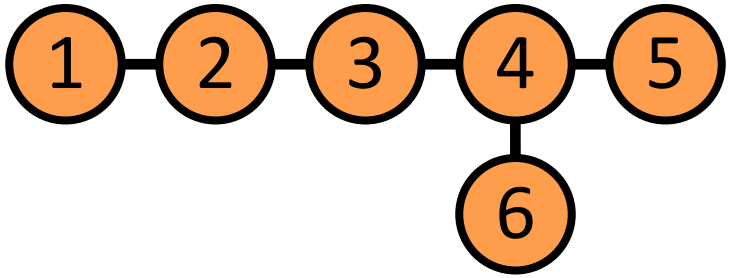
- Knoten: $V = \{v_1,\ldots,v_n\}$
- $A \in \text{Bool}^{n \times n}$
- $a_{ij} = 1$ falls $(i,j) \in E$, $a_{ij} = 0$ sonst
- j-te Spalte: alle Vorgänger von v_j
- i-te Zeile: alle Nachfolger von v_i
- A symmetrisch für ungerichtete Graphen

Adjazenzmatrix



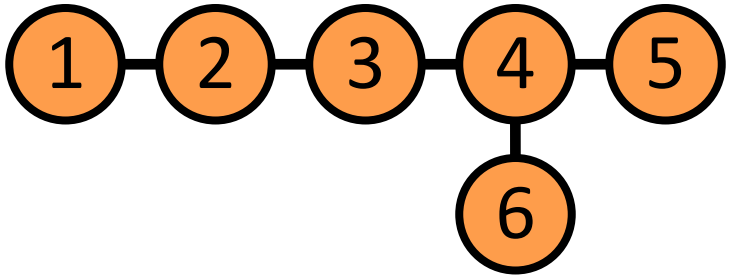
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Zusammenhangskomponenten



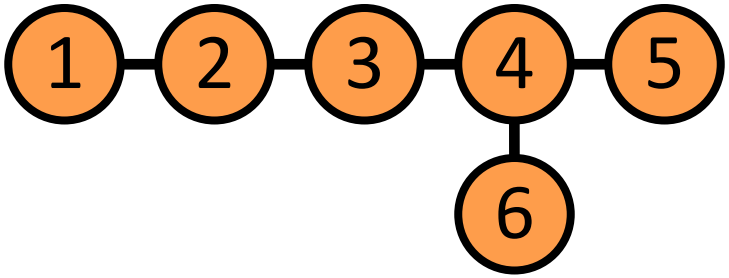
$$B = A + I = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Zusammenhangskomponenten



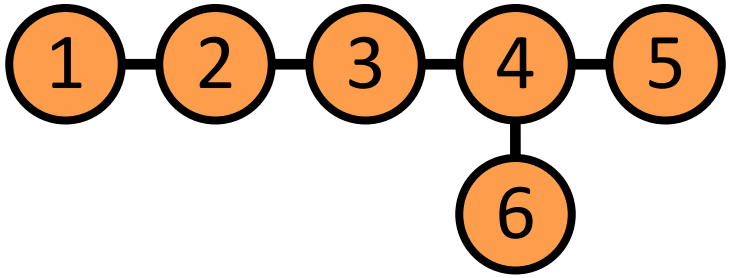
$$B^2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Zusammenhangskomponenten



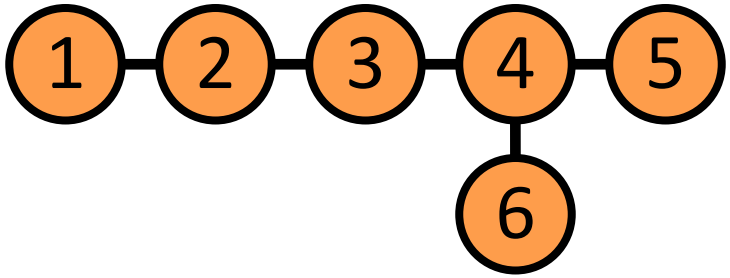
$$B^3 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Zusammenhangskomponenten



$$B^4 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Zusammenhangskomponenten



$B^4 =$

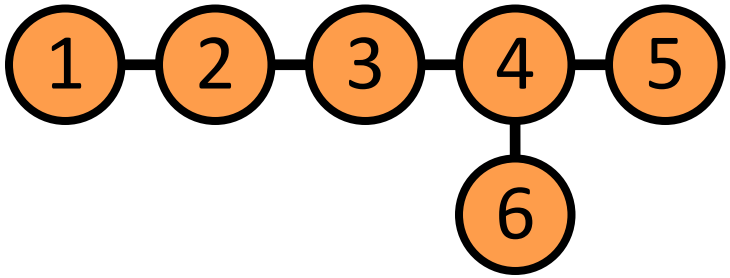
$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Durchmesser

Adjazenzlisten

- Bei komplexen Graphen bestehen oft nur lokale Knotenbeziehungen
- Die meisten Einträge der Adjazenzmatrix sind „0“ (dünn besetzte Matrix)
- Speichere nur die existierenden Kanten

Adjazenzlisten



1: 2
2: 1,3
3: 2,4
4: 3,5,6
5: 4
6: 4

Adjazenzlisten

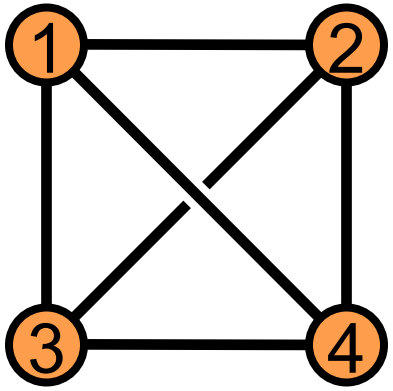
- Für gerichtete Graphen werden Vorgänger- und Nachfolger-Listen verwaltet.
- Speicherverbrauch: (#Nachbarn ≤ k)
  - A-Matrix: n^2 bit
  - A-Listen: $n \times k \times b = n \times k \times \log n$
- Beispiel: Regionalbahnen in Aachen und Wladiwostok

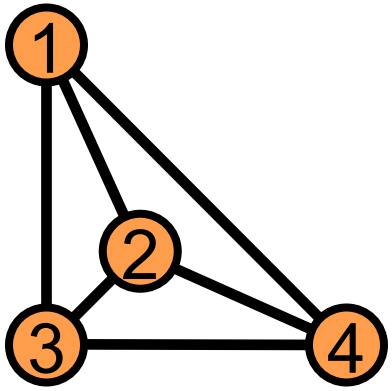
Planare Graphen

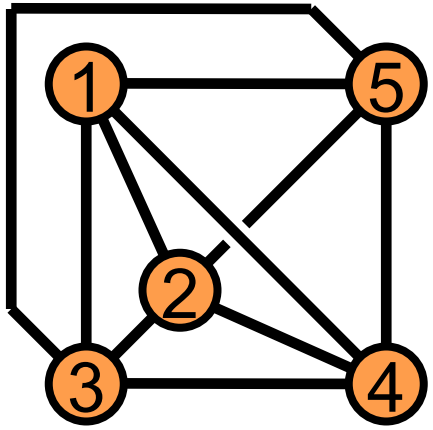
- ... beschreiben eine Partition der Ebene in disjunkte Zellen.
- Knoten sind durch Kanten verbunden und diese bilden Zellen (Maschen).
- Planar: es ist möglich die Knoten so zu platzieren, dass sich die Kanten nicht kreuzen.

Planare Graphen

- Kompatibilitätsbedingungen
  - eindeutiger Umlaufsinn für die Nachbarn jedes Knotens
  - eindeutiger Umlaufsinn für die Kanten jeder Zelle / Masche

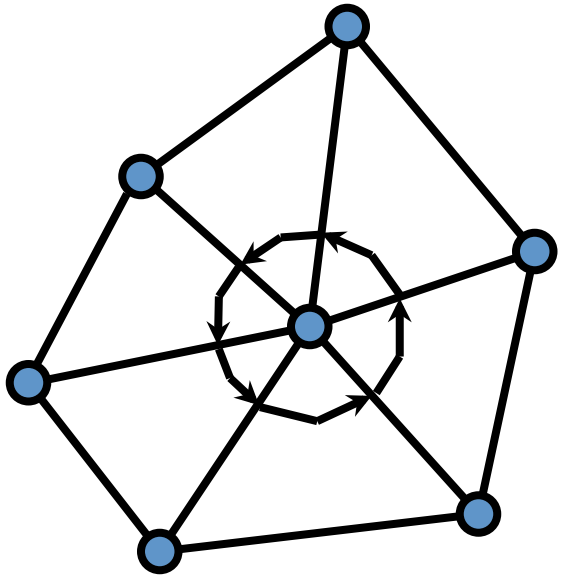






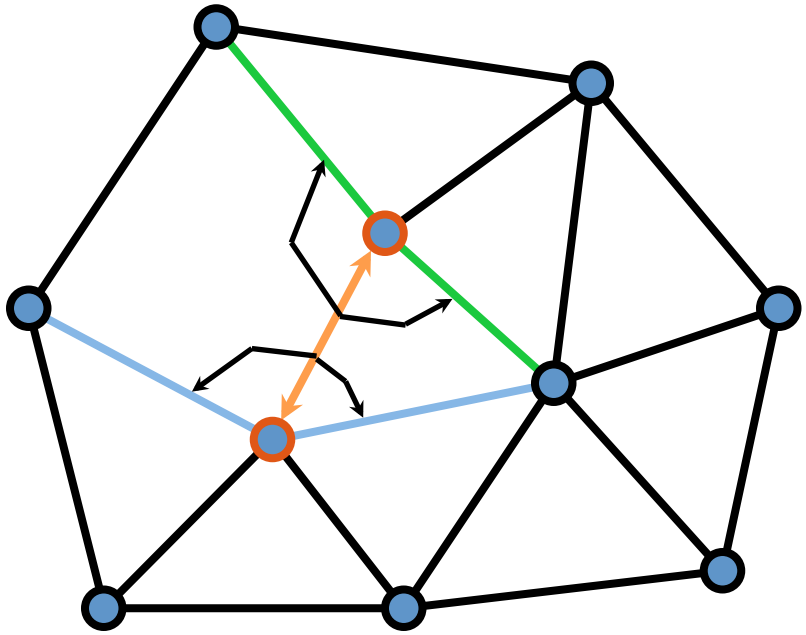
Kantenstrukturen

- Kanten verbinden Knoten
- Kanten verweisen auf benachbarte Kanten (Umlaufsinn)
- Garantiert die Planarität



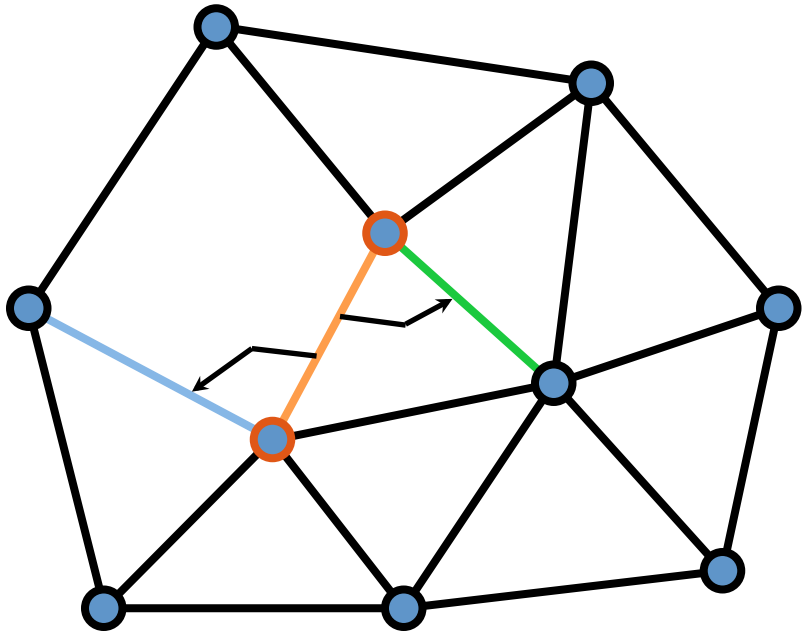
Kantenstrukturen

- class Edge {
  Node A,B
  Edge N_1,N_2
  Edge P_1,P_2
  }



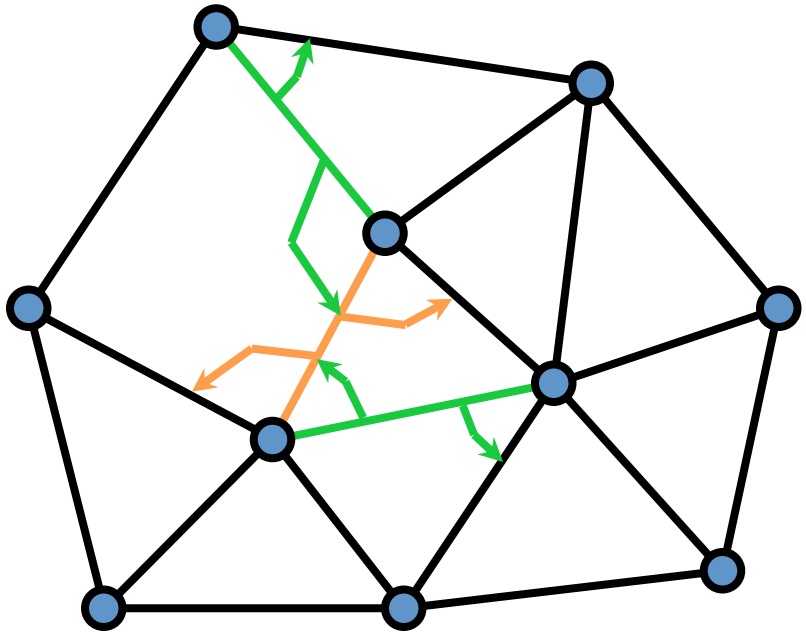
Kantenstrukturen

- class Edge {
  Node A,B
  Edge Next
  Edge Prev
  }



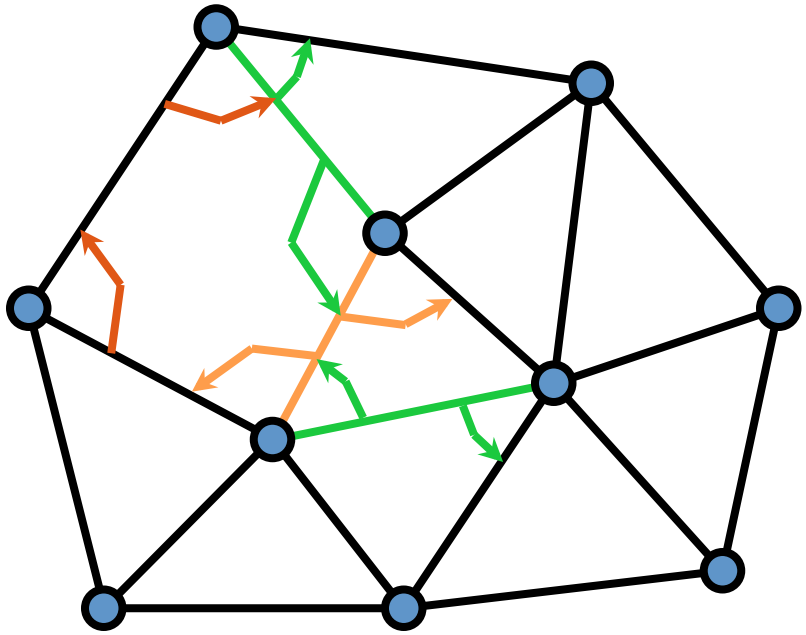
Kantenstrukturen

- class Edge {
  Node A,B
  Edge Next
  Edge Prev
  }



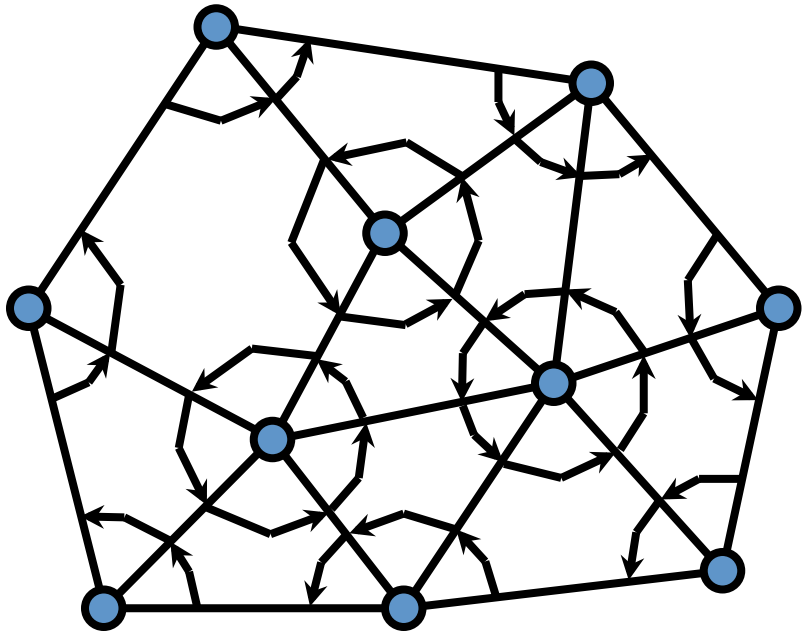
Kantenstrukturen

- class Edge {
  Node A,B
  Edge Next
  Edge Prev
  }



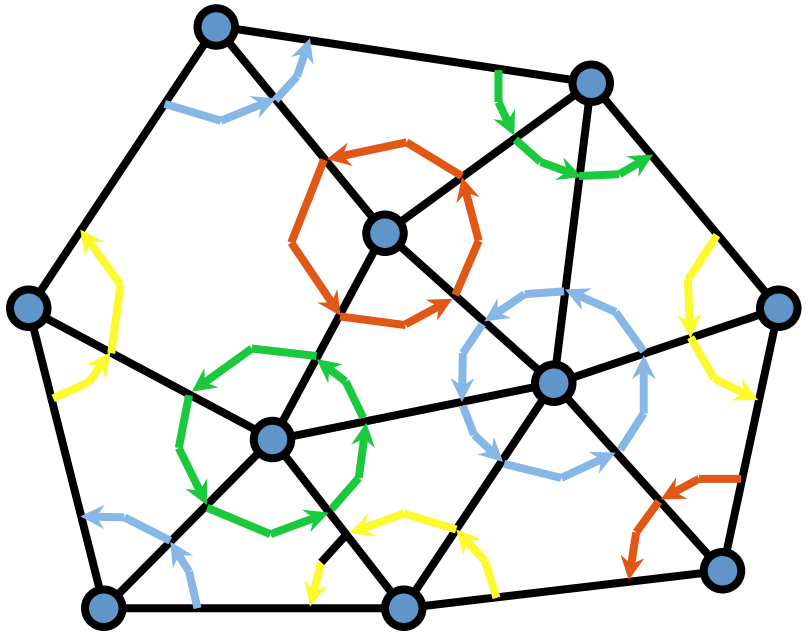
Kantenstrukturen

- class Edge {
  Node A,B
  Edge Next
  Edge Prev
  }



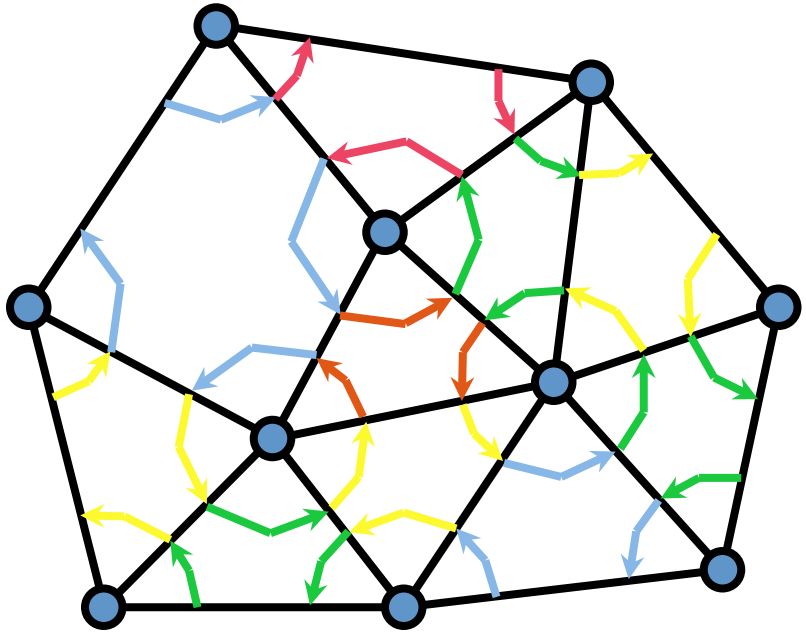
Kantenstrukturen

- class Edge {
  Node A,B
  Edge Next
  Edge Prev
  }

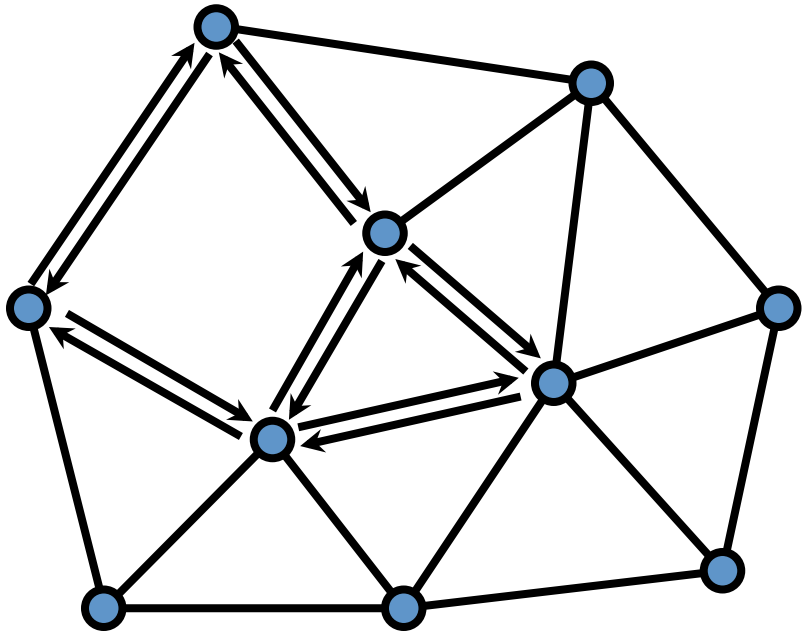


Kantenstrukturen

- class Edge {
  Node A,B
  Edge Next
  Edge Prev
  }

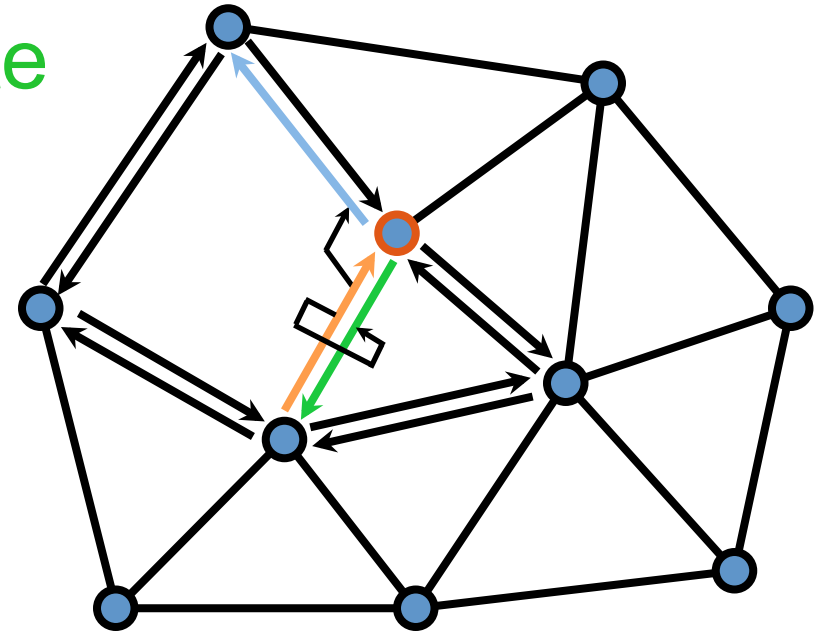


Halbkantenstrukturen



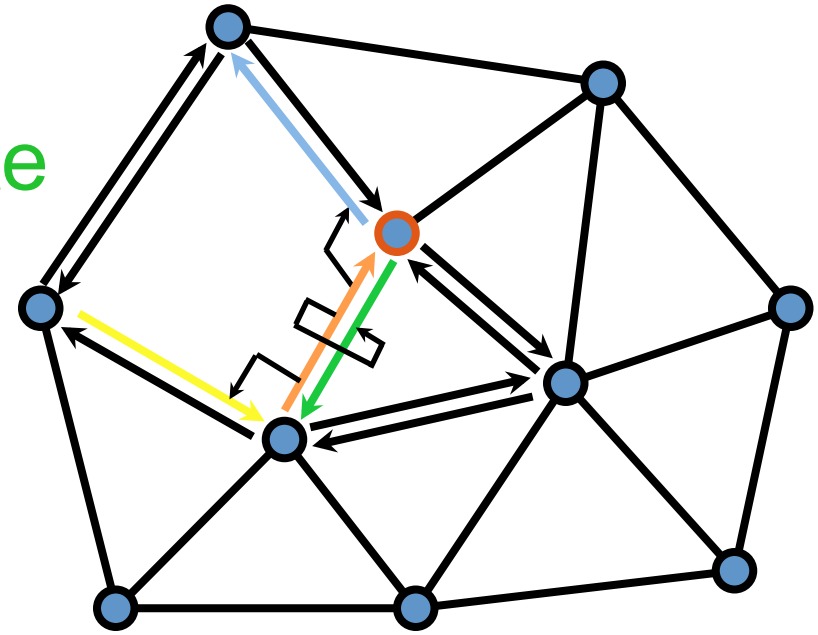
Halbkantenstrukturen

- class HEdge {
  Node A
  Edge Next
  Edge Opposite
  }



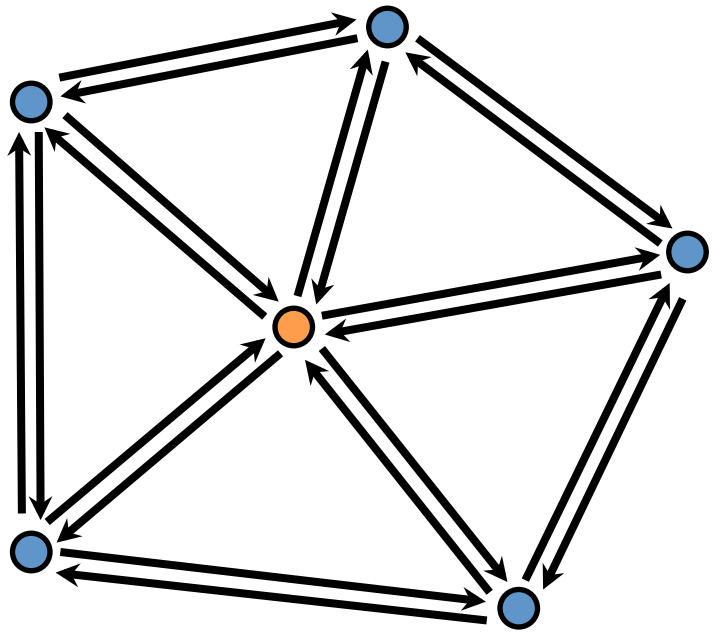
Halbkantenstrukturen

- class HEdge {
  Node A
  Edge Next
  Edge Prev
  Edge Opposite
  }



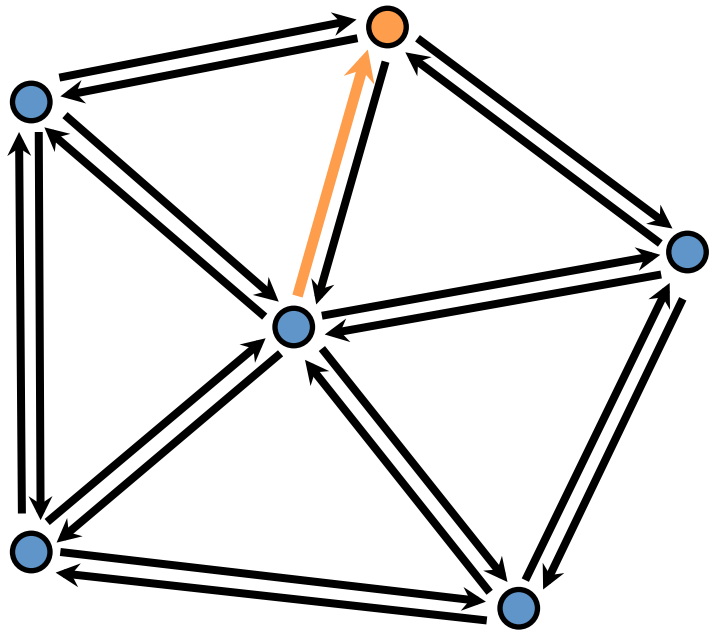
Liste der Nachbarn

1. Start at vertex



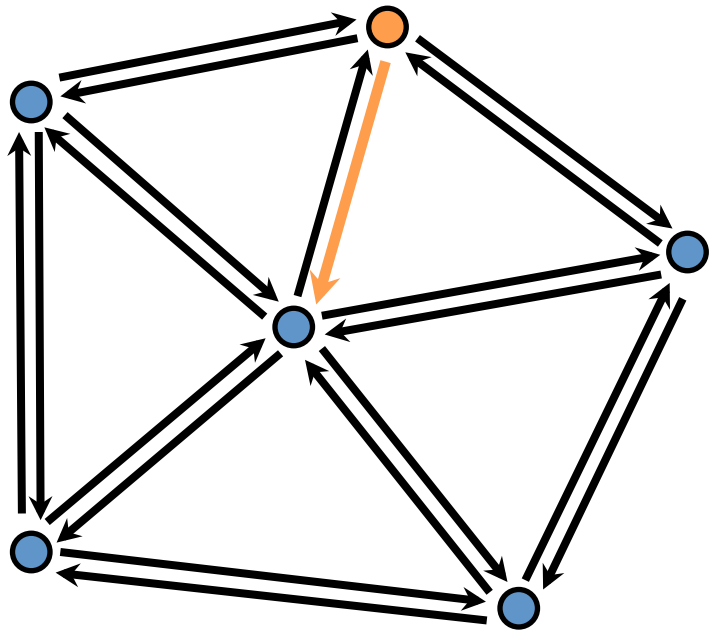
Liste der Nachbarn

1. Start at vertex
2. Outgoing halfedge



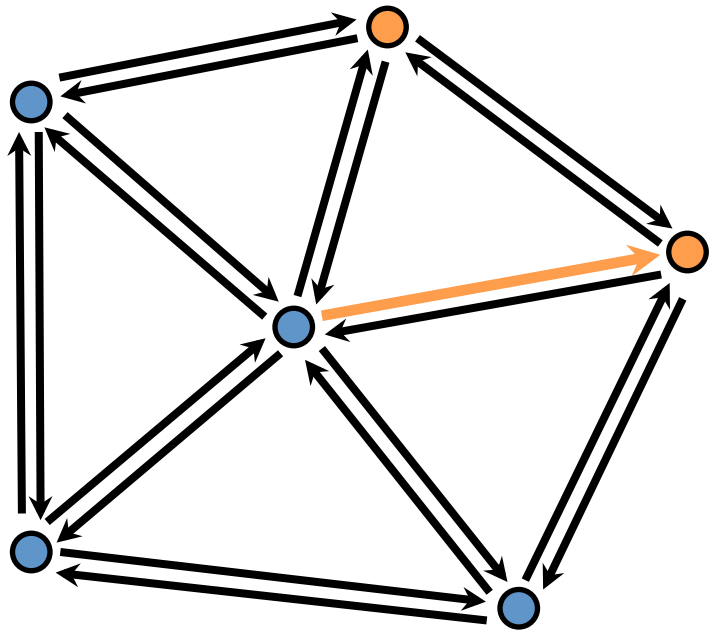
Liste der Nachbarn

1. Start at vertex
2. Outgoing halfedge
3. Opposite halfedge



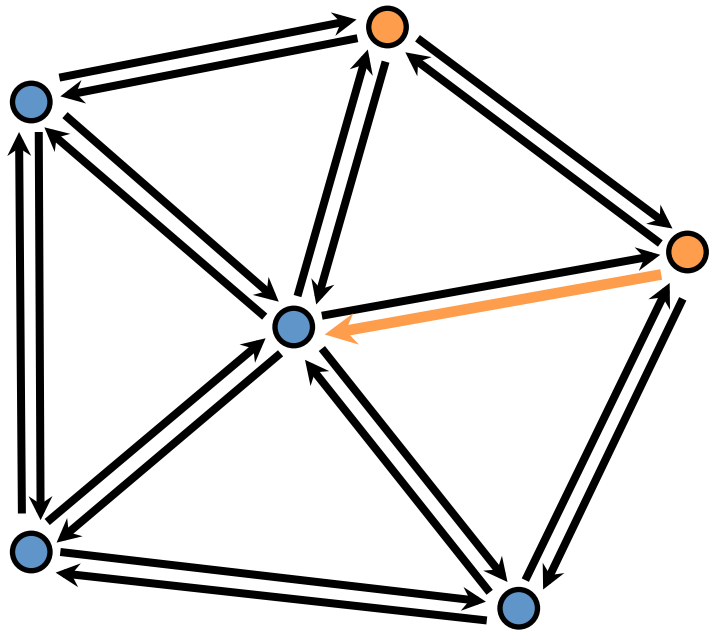
Liste der Nachbarn

1. Start at vertex
2. Outgoing halfedge
3. Opposite halfedge
4. Next halfedge



Liste der Nachbarn

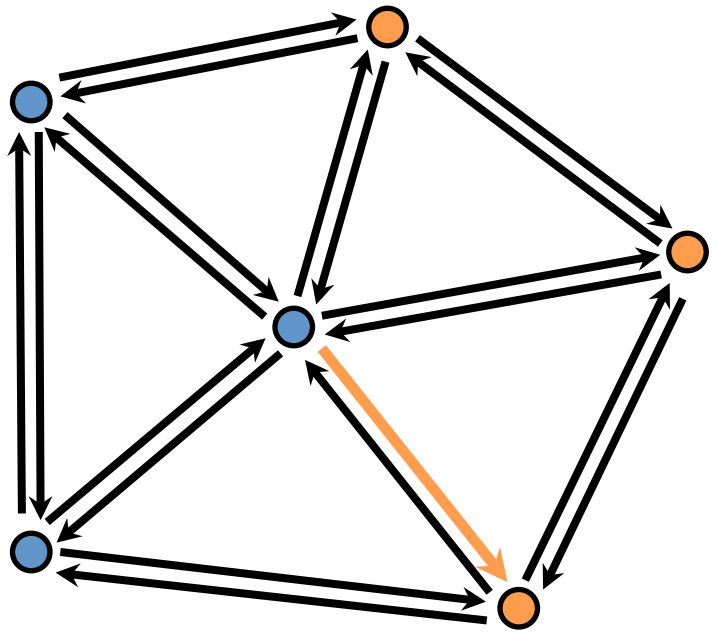
1. Start at vertex
2. Outgoing halfedge
3. Opposite halfedge
4. Next halfedge
5. Opposite halfedge



Liste der Nachbarn

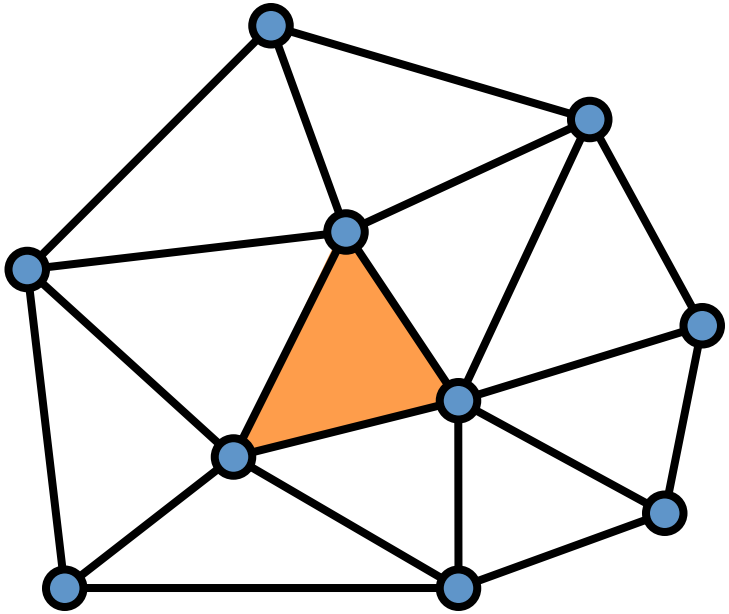
1. Start at vertex
2. Outgoing halfedge
3. Opposite halfedge
4. Next halfedge
5. Opposite halfedge
6. Next halfedge

...



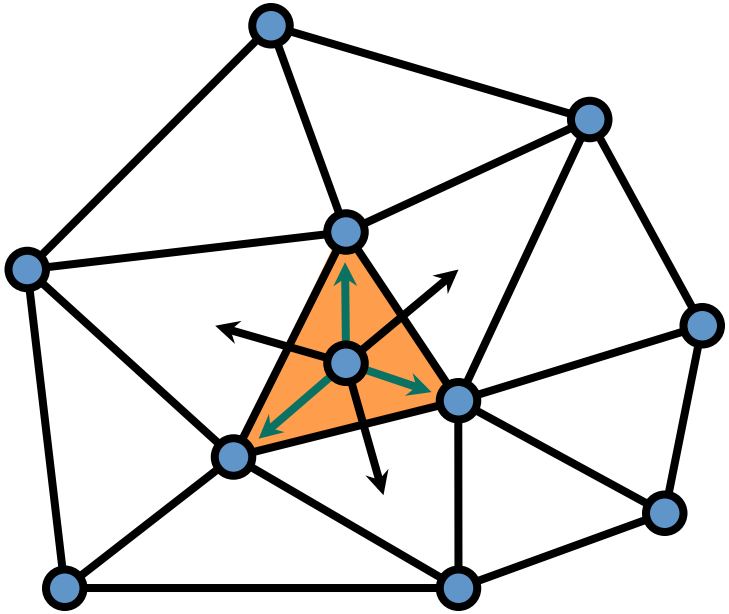
Simplex-Strukturen

- Simplices sind die einfachsten k-dim. Strukturen
  - Punkt
  - Strecke
  - Dreieck
  - Tetraeder
  - ...



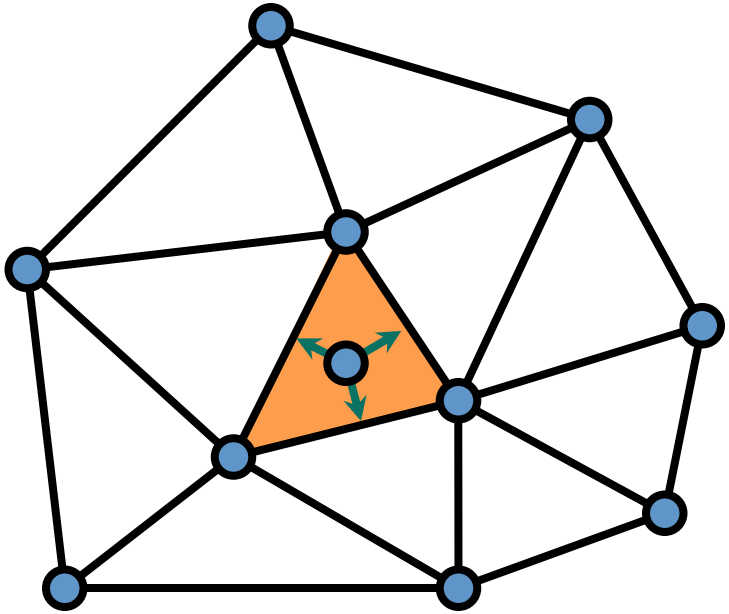
Simplex-Strukturen

- Ein k-Simplex hat
  - k+1 Knoten
  - k+1 Nachbarn



Simplex-Strukturen

- Seine k+1 Flächen sind (k−1)-Simplices



Simplex-Strukturen

- class k_Simplex {
  Node Vertices[1..k+1]
  (k−1)_Simplex Faces[1..k+1]
  }
- Darstellung beliebig dimensionaler Komplexe mit disjunkten Zellen
- Navigation / Aufzählung schwieriger