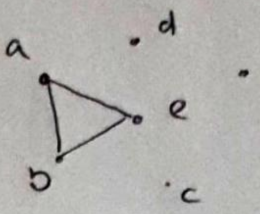
2 Grundlagen und Begriffe:

V = knoten | V^2 = kanten = E



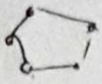
Ein Graph - (V,E) mit $V = \{a,b,c,d,e\}$

$E - \{\{a,b\},\{e,a\}$ ....

p = knoten | q = kanten

$\Delta(G)$ = Grad = Anzahl der kanten an einem Knoten

---

kantenfolge: geschlossen =  $V_0 = V_n$ | Kreis

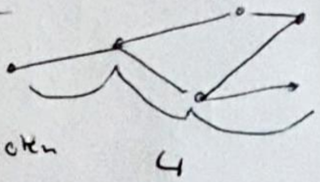
kantenzug =  | Weg

---

Auf spannender Graph = Wenn knoten identisch

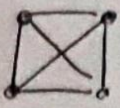
induzierter Graph = nicht alle Knoten

Abstand ist die Läng der kürzisten Weges



Durchmesser ist der ~~Abstand~~ größt Abstand aller knoten

3.1 Vollständige Graphen

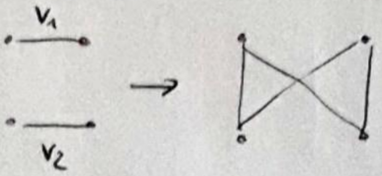


jeder knoten ist miteinander verbunden.

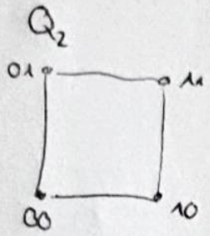
Kantenzahl = $\frac{n(n-1)}{2}$

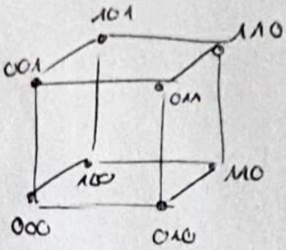
3.2 K-prtite Graphen

Jeder Punk einer Menge hat eine Kante zu einem Knoten der anderen Menge



Hypercube





Vollständiger K-patiter Graph

Jeder Punkt V_i ist mit allen Knoten in V_j verbunden

3.3 Reguläre Graphen

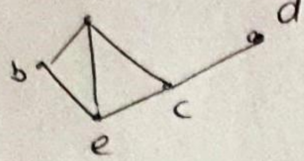
$\Delta(G) = \delta(G)$ Maximaler Grad des Graphen

= alle Grade der Graphen

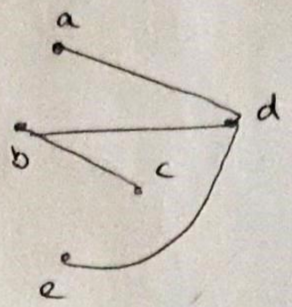
3.4 Komplement eines Graphen

$V(\bar{G}) - V(G)$

G:



$\bar{G}$:



4. Besondere Wege und Kreise

- Eulerische Linie - jede Kante von G max und min 1 mal
- Kreis: geschlossene Linie
- Graph: Eulerischer Kreis → Graph

4.2 Hamiltonsche Kreise

- ein Kreis C in G der alle Knoten beinhaltet.

5 Planare Graphen

Kanten dürfen sich nicht überschneiden

Sei p=Knoten, q=Kanten, f=Flächen

$$p+f=q+2$$

$$q \geq \frac{3 \cdot f}{2}$$

$$\Rightarrow 10 \geq \frac{3f}{2}$$! ↯

6 Bäume

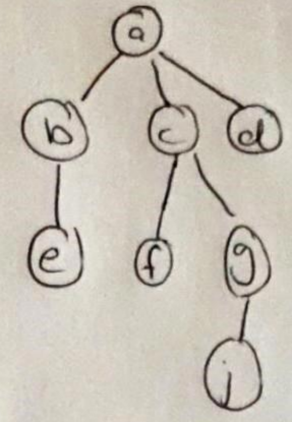
Graph ohne Kreis = Wald

Teilgraph = Baum

$T \Rightarrow p \geq 2$ → min. 2 Knoten vom Grad 1

n Knoten → $n-1$ Kanten

6.2 Wurzelbäume



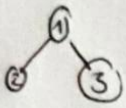
a ist eine **Wurzel**

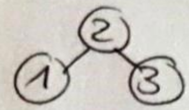
e, f, j, d ist ein **Blatt**

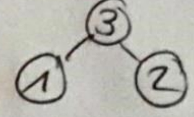
b, c, g **innere Knoten**

c ist **Verfahre** von f. g

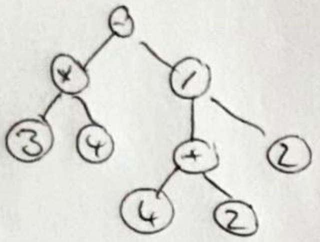
b, c, d **Geschwister**

Preorder: 

Inorder: 

Postorder: 

Inorder:



Postorder:

3·4 − (4+2) / 2

3 4 · 4 2 + 2 / −

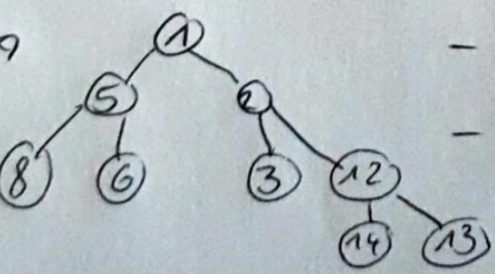
6.3 Suchbäume

Linker Nachfahre muss kleiner sein als Vater
rechter " —— " größer " —— "

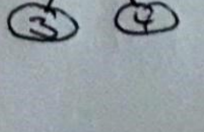
Suche im Suchbaum, if m > als Knoten nach rechts
else nach links

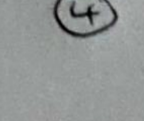
6.4 Heap (Haufe)

Kl=max

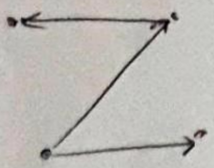


- Kleinste Makierung immer oben
- Knoten von Links nach rechts





7 Gerichtete Graphen



Graph mit Pfeilen bzw. Bögen

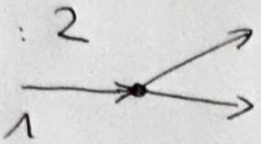
→ es gibt gerichtete Wege und Kreise

Schwach zusammenhängend: G ist zusammenhängend

Stark zusammenhängend: Jeder Knoten ist von jedem erreichbar

↳ starker Teilgraph = Starke Komponente

Außengrad: 2



Innengrad: 1

$\text{grad } v = (\text{grad}^+ v) + (\text{grad}^- v)$

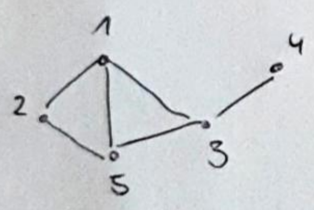
If Innengrad = 0 ⇒ Quelle

If Außengrad = 0 ⇒ Senke

8. Graphen und Matrizen

8.1. Adjazenzmatrix eines Graphen (A)

Wenn es eine Kante gibt 1, sonst 0



$$A^1 = \begin{array}{c|ccccc|l} & & & \text{nach} & & & \\ 1 & 0 & 1 & 1 & 0 & 1 & \\ 2 & 1 & 0 & 0 & 0 & 1 & \\ 3 & 1 & 0 & 0 & 1 & 1 & \text{von} \\ 4 & 0 & 0 & 1 & 0 & 0 & \\ 5 & 1 & 1 & 1 & 0 & 0 & \\ & 1 & 2 & 3 & 4 & 5 & \end{array}$$

Bei gerichtet nur in eine Richtung

A^L ist die Anzahl der Schrittgröße

$$A^2: \begin{array}{c|ccccc|} & \cancel{1} & \cancel{2} & \cancel{3} & \cancel{4} & \cancel{5} \\ \cancel{1} & 3 & 1 & 1 & 1 & 2 \\ \cancel{2} & 1 & 2 & 2 & 0 & 1 \\ \cancel{3} & 1 & 2 & 3 & 0 & 1 \\ \cancel{4} & 1 & 0 & 0 & 1 & 1 \\ \cancel{5} & 2 & 1 & 1 & 1 & 3 \end{array}$$

8.3 Erreichbarkeit- / Wegmatrix

W(G) ist die Wegmatrix vom Graphen G

$$w_{i,j} = \begin{cases} 1 & \text{wenn Weg (Kante) vorhanden} \\ 0 & \text{wenn nicht} \end{cases}$$

Warshall - Alg

beginnt $w_{i,j}^{(0)} = a_{i,j}$

$$\rightsquigarrow w_{i,j}^{(k)} = \begin{cases} 1: & \text{falls } w_{ij}^{(k-1)} = 1 \text{ (Es gab vorher schon einen Weg} \\ 1: & \text{falls } w_{ik}^{(k-1)} = 1 \text{ und } \uparrow \text{ (Umweg über k)} \\ 0: & \text{sonst} \end{cases}$$

$W^{(0)}$ = Erreichbarkeit ab start

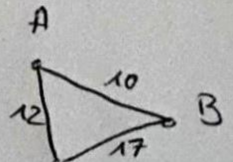
$W^{(1)}$ = Erreichbarkeit von $W^{(0)}$ + 1ne Kante

$$W^{(0)} \begin{array}{c} \\ \\ y \\ \\ z \end{array} \overset{\displaystyle \begin{array}{ccccc} & x & & & \end{array}}{\left(\begin{array}{c|cccc} 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{array}\right)} \xRightarrow{k=1} W^{(1)} \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & & & & \\ 1 & ① & & & \\ 0 & & & & \\ 1 & ① & & & \end{pmatrix}$$

Weil x mit y und z überschneidet 1 haben

Erreichbarkeitsmatrix = transitiver Abschluss des Graphen

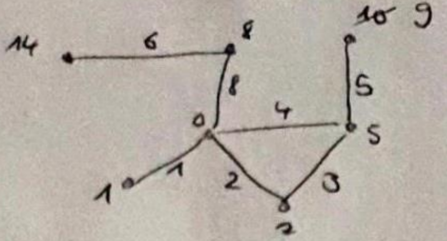
9. Bewertete Graphen



Kanten bekommen Werte

Kantenfolge = Summe aller Werte der Kanten von A → C

Bsp. (10 + 17) v (12)



9.1. Kürzester Weg

Floyd-Warshall

Starte $d_{ij}^{(0)} = \begin{cases} 0 & \text{falls } i = j \\ w_{ij} & \text{falls } i \neq j \text{ und } a_{ij} = 1 \\ \infty & \text{sonst} \end{cases}$

$D = \text{Distanzmatrix} = D^{(n)} = d_{ij}^{(n)}$

Durchmesser $= \max. D$ von $(G) \rightsquigarrow d(G) = \max d_{ij}$

Bsp: $A(G) \rightsquigarrow D^{(0)}$

$$A(G) \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix} \rightsquigarrow D^{(0)} \begin{pmatrix} 0 & 1 & 1 & \infty & 1 \\ 1 & 0 & \infty & \infty & 1 \\ 1 & \infty & 0 & 1 & 1 \\ \infty & \infty & 1 & 0 & \infty \\ 1 & 1 & 1 & \infty & 0 \end{pmatrix} \overset{k=1}{\Rightarrow} D^{(1)} \begin{pmatrix} 0 & 1 & 1 & \infty & 1 \\ 1 & 0 & 2 & \infty & 1 \\ 1 & 2 & 0 & 1 & 1 \\ \infty & \infty & 1 & 0 & \infty \\ 1 & 1 & 1 & \infty & 0 \end{pmatrix}$$

2 weil
$1 + 1 = 2$

$$\overset{k=2}{\Rightarrow} D(2) \begin{pmatrix} 0 & 1 & 1 & \infty & 1 \\ 1 & 0 & 2 & \infty & 1 \\ 1 & 2 & 0 & 1 & 1 \\ \infty & \infty & 1 & 0 & \infty \\ 1 & 1 & 1 & \infty & 0 \end{pmatrix} \overset{k=3}{\Rightarrow} D(3) \begin{pmatrix} 0 & 1 & 1 & 2 & 1 \\ 1 & 0 & 2 & \textcircled{3} & 1 \\ 2 & \textcircled{2} & 1 & 0 & 1 \\ 1 & 2 & 0 & 1 & 1 \\ 1 & 1 & 1 & 2 & 0 \end{pmatrix}$$

Durchmesser: $d(G) = \max d_{ij}^{(5)} = 3$

9.2 Minimale spannende Bäume

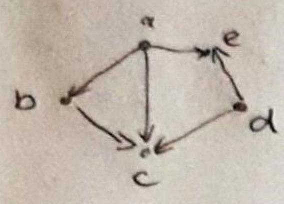
Spannender Baum mit min. Summe der Wertung (Kosten)

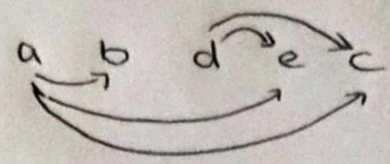
Algoritmus von Prim:
- Starte an einem beliebigen Punkt
- ziehe Kante mit geringster Kosten
- ziehe Kante mit geringten Kosten von irgendeinen Knoten der bereit im Baum ist.

Algoritmus von Kruskal:
- bis alle Knoten im Baum sind
- ziehe Kante mit geringsten Kosten
- ziehe nächste mit nächst günstign Koste
- kein Kreis
- bis alle Knoten im Baum enthalten

10 Topologisches Sortieren

ist gerichtet, kein Kreis





Man nimmt sich eine Quelle, entfert alle Kanten, nimmt die nächste Quelle die daus folgt bis alle Knoten stehen und verbindet diese wie im Graphen.

11. Suchstrategien für Graphen

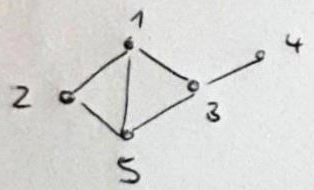
BFS: **Breitensuche**: zuerst erreichte Knoten werden bearbeitet

DFS: **Tiefensuche**: zuletzt erreichte Knoten werden bearbeitet.

11.1 Breitensuche

Speichert Knoten in Reihenfolge des erreichens. Neue Knoten hinten, bearbeitende Knoten vorne.

White: none | Gray: in Bearbeitung | Black: finished



Start 2

=> 2 | 1,5 | 5,3 | 3 | 4 | ∅

=> Reihenfolge = 2,1,5,3,4

11.2 Tiefensuche

Speichert Knoten in der Reihenfolge des erreichens auf Stack

White: none | Gray: in Bearbeitung | Black: finished

Start 2:

=> 2 | 2,1,5 | 2,1,5,1,3 | 2,1,5,1,3,1,4 |
2,1,5,1,3,1,4 | 2,1,5,1,3,1 | 2,1,5,1,3 |
2,1,5,1 | 2,1 | 2 | ∅

=> Knotenreihnfolge: 4,1,3,5,2

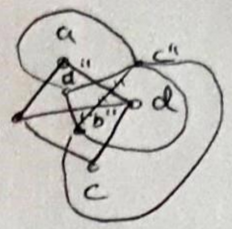
11.3 Tiefensuche in gerichteten Graphen

Tiefensuche wird auf sämtliche Knoten als S angewendet

12. Duale Graphen

Andere Darstellung planarer Graphen

12.1 Dualer Graph



- Man habe einen Planaren Graphen

→ in jede Fläche kommt ein Knoten"

→ jeder 2 Knoten" die in benachbarten Flächen liegen erhalten eine Kante

Planarer Graph $G = p \,,\, f \,;\, q$

dualer Graph $G^* = \overset{''}{f^*},\ \overset{''}{p^*}\ \overset{''}{;}\ q^*$

12.2 Platonische Graphen

Graph mit besonders „gleichmäßigen" Aufbau

m	n	$q = \frac{2mn}{2m+2n-mn}$	$p = \frac{2q}{m}$	$f = \frac{2q}{n}$	Namen der Graphen
3	3	6	4	4	Tetra
3	4	12	8	6	Hexa
3	5	30	20	12	Dode
4	3	12	6	8	Okta
5	3	30	12	20	Ikosa

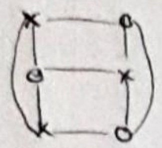
- Hexa ist dual zu Okta
- Dode ist dual zu Ikosa
- Tetra zu Tetra

13 Färbungen von Graphen

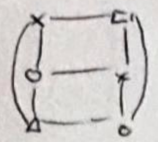
Knoten werden gefärbt. Farben werden vorher festgelegt.

- K-Farben → Name K-Färbung
- zulässig = nicht die selbe Farbe bei 2 benachbarten Knoten
- chomatische Zahl $x(G)$ = niedrigste zulässige K-Färbung

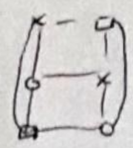
x = rot o = blau □ = grün △ = türkis



2-Färbung



zulässige 4-Färbung



zulässige 3-Färbung

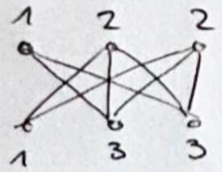
Bei dem G ist $x(G) = 3$

- Pseudographe besitzen keine zulässige Färbung (Schlingen)
- Mehrfachgraphen egal (Kein Einfluss)

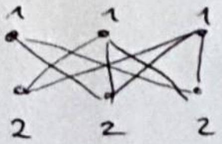
13.1 Allgemeine Aussagen

- bei vollständigen Graphen $K_n \Rightarrow x(K_n) = n$

⇝ Farben werden als natürliche Zahlen dargestellt.



} 3-Färbung



} zulässig 2-Färbung

- Jeder k-patiter Graph lässt sich ~~als~~ zulässig färben
  - bipartit → zulässig 2-Farben
  - vollständig bipartit → $x(K_{m,n}) = 2$
  - Kreise gerade Länge → $x(C_{2n}) = 2$
  - Kreise ungerade Länge → $x(C_{2n+1}) - 3$
  - jeder nicht Triviale Baum T → $x(T) = 2$