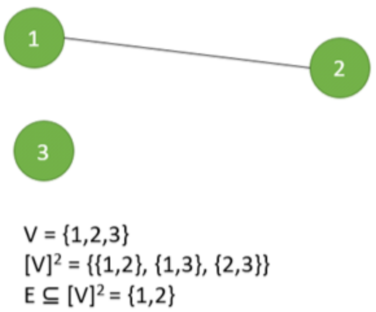
Fragenkatalog AMS4VO_BB

1. Was ist ein Graph?

Ein Graph besteht aus einer Menge von Knoten (auch als "Vertices" oder "Ecken" bezeichnet) und einer Menge von Kanten, die diese Knoten verbinden. Ein Graph wird oft durch eine Diagrammdarstellung dargestellt, bei der die Knoten als Punkte und die Kanten als Linien oder Pfeile dargestellt werden.



2. Welche Arten von Graphen kennen Sie?

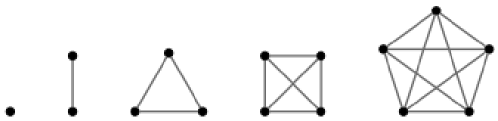
- `ungerichter Graph`
    - Ein (ungerichteter) Graph (engl. undirected graph) G ist ein 2-Tupel
    - G = (V, E)
      1. V *endliche Menge*
      2. $E \subseteq [V]^2$
         - $e = \{u, v\},$
    - Reihenfolge der Tupel ist egal
    - V heißt Knotenmenge, die Elemente aus V sind Knoten (engl. vertices).
      E ist die Kantenmenge und deren Elemente heißen Kanten
      Die Kante e verbindet die Knoten u und v (engl. edges).
- `gerichteter Graph`
- Ein gerichteter Graph G ist ein Tupel G = (V,E) mit:
    - 1. V, E *zwei endliche Mengen.*
    - 2. $E \subseteq V \times V$, *das heißt für* $e \in E$*:*
         - $e = (u, v)$, *mit* $u \neq v,\ u, v \in V$ *oder*
         - $e = (v, v)$, *mit* $v \in V$. *Die Kante ist also eine Schleife.*
- Eine Kante e = (u, v) bedeutet, dass die Knoten u und v durch die gerichtete Kante e verbunden werden, wobei u als Startknoten und v als Endknoten der Kante bezeichnet wird
- `multigraph`

- Ein ungerichteter allgemeiner Multigraph G ist ein Tripel G = (V, E, i) mit
  1. *V, E zwei endliche Mengen.*
  2. *$i\colon E \to [V]^2 \cup [V]^1$, das heißt für $e \in E$:*
     - *$i(e) = \{u, v\}$, mit $u \neq v$, $u, v \in V$ oder*
     - *$i(e) = \{v\}$, mit $v \in V$.*
- Schleife: i(e) = {v}
- Falls i injektiv ist (keine Mehrfachkanten) und es keine Schleifen gibt (also i: E --> [V]^2) dann spricht man von schlichten Graphen (simple graph)

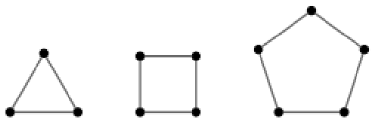
* gewichtet

Exkurs wichtige spezielle Graphen

1. **Der vollständige Graph K_n mit n Knoten** (engl. complete graph)
   $V = \{1, \ldots, n\}$, $E = [V]^2$



2. **Der n-Zyklus C_n**
   $V = \{1, \ldots, n\}, n \geq 3$, $E = \{\{i, i+1\} : 1 \leq i \leq n-1\} \cup \{\{1, n\}\}$



3. **Der n-Pfad P_n**
   $V = \{0, \ldots, n\}$, $E = \{\{i, i+1\} : 0 \leq i \leq n-1\}$

3. Wie unterscheiden sich gerichtete von ungerichteten Graphen in der Definition?

- Ein ungerichteter Graph G ist ein 2-Tupel G = (V, E) mit
  - 1. *V endliche Menge*
    2. *$E \subseteq [V]^2$*
       - *$e = \{u, v\}$,*

- Ein gerichteter Graph G ist ein Tupel G = (V, E) mit
  - 1. *V, E zwei endliche Mengen.*
  - 2. *$E \subseteq V \times V$, das heißt für $e \in E$:*
       - *$e = (u, v)$, mit $u \neq v$, $u, v \in V$ oder*
       - *$e = (v, v)$, mit $v \in V$. Die Kante ist also eine Schleife.*

4. Wie werden Graphen dargestellt?

`Knotendiagramm/grafische Darstellung`: In einem Knotendiagramm werden die Knoten eines Graphen als Punkte dargestellt, während die Kanten als Linien oder Pfeile zwischen den Punkten dargestellt werden.

Können entweder gerichtet sein, bei denen die Kanten Pfeile haben, oder ungerichtet, bei denen die Kanten Linien sind.

`Adjazenzliste`:
Vor allem wenn ein Graph relativ wenig Kanten im Vergleich zur Anzahl der Knoten hat

Sei $G = (V, E)$ ein Graph mit Knoten $v_1, \dots, v_n$ und Kanten $e_1, \dots, e_m$. Dann heißt der Vektor $L = (L_i)_{i=1,\dots,n}$ Adjazenzliste *(engl. adjacency list), wobei*

$$L_i = \{v \in V \mid v \sim v_i\}\ .$$

Für gerichtete Graphen werden statt der Nachbarn die Nachfolger in die Liste geschrieben.

Beispiel 1.32.
Adjazenzliste für einen einfachen Graph:

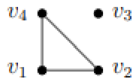
$$(\{v_2, v_4\}, \{v_1, v_4\}, \{\}, \{v_1, v_2\})$$

`Adjazenzmatrix`: Eine Adjazenzmatrix ist eine quadratische Matrix, in der die Zeilen und Spalten den Knoten des Graphen entsprechen. Die Elemente der Matrix geben an, ob eine Kante zwischen den Knoten besteht oder nicht. Eine "1" oder ein anderer Wert wird verwendet, um eine Kante anzuzeigen, während eine "0" oder ein leerer Wert angezeigt wird, wenn keine Kante vorhanden ist.

Sei $G = (V, E)$ ein Graph mit Knoten $v_1, \dots, v_n$ und Kanten $e_1, \dots, e_m$. Dann heißt die $(n \times n)$-Matrix $A = (a_{ij})_{i,j=1,\dots,n}$ mit

$$a_{ij} = \begin{cases} 1 & \textit{falls } \{v_i, v_j\} \in E \\ 0 & \textit{sonst} \end{cases}$$

eine Adjazenzmatrix *(engl. adjacency matrix). Für gerichtete Graphen ist die Matrix nicht symmetrisch.*



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

Adjazenzmatrizen von ungerichteten Graphen sind immer symmetrisch und bei einfachen Graphen sind die Einträge in der Hauptdiagonale alle Null.
Zeilen- und Spaltensummen entsprechen den Knotengraden der jeweiligen Knoten.

`Inzidenzmatrix`:

Sei $G = (V, E)$ ein Graph mit Knoten $v_1, \ldots, v_n$ und Kanten $e_1, \ldots, e_m$. Dann heißt die $(n \times m)$-Matrix $B = (b_{ij})_{\substack{i=1,\ldots,n \\ j=1,\ldots,m}}$ mit

$$b_{ij} = \begin{cases} 1 & \textit{falls } v_i \textit{ liegt auf } e_j \\ 0 & \textit{sonst} \end{cases}$$

eine Inzidenzmatrix (engl. incidence matrix).

Sei $B^\top$ die transponierte Matrix zu B. Dann gilt für die $(n \times n)$ Matrix $M = BB^\top = (m_{i,j})_{i,j=1,\ldots,n}$, dass

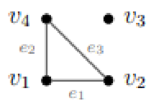
$$m_{ij} = \begin{cases} d(v_i) & \text{falls } i = j \\ a_{ij} & \text{sonst.} \end{cases}$$

Also:

$$M = \begin{pmatrix} d(v_1) & 0 & \cdots & 0 \\ 0 & d(v_2) & 0 & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & \cdots & d(v_n) \end{pmatrix} + A,$$

wobei $A = (a_{ij})_{i,j=1,\ldots,n}$ die Adjazenzmatrix ist.

Beispiel 1.30.
Inzidenzmatrix für einen einfachen Graph:



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

5. Was ist ein gewichteter Graph?

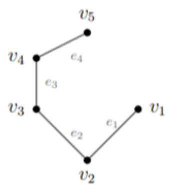
- Kanten erhalten Gewicht
- Gewichtsfunktion w weist jeder Kante Zahl zu
- Kann sowohl bei ungerichtetem als auch gerichtetem Graph vorkommen

6. Wie wird das Kantengewicht mathematisch definiert?

- Über eine Gewichtungsfunktion ω: E -> ℕ
- Gewichtsfunktion = Funktion, die jede Kante in einem Zahlenraum abbildet
- G = (V,E)
- Paar (G, ω) ist ein gerichteter Graph

7. Was ist Nachbarschaft in Graphen?

- Zwei Knoten u, v ∈ V mit u ≠ v heißen `benachbart` (adjazent, engl. adjacent), falls es eine Kante gibt, die u und v verbindet.
    - `benachbart/adjazent`: u ~ v
    - u ~ v := ∃ e∈E ∧ u∈e ∧ v∈e ∧ u≠v
- Zwei Kanten e1, e2 mit e1 != e2 heißen benachbart (`inzident`, engl. incident), falls es einen Knoten gibt, der auf beiden Kanten liegt
    - e1 ~ e2 := ∃ v∈V ∧ e1∈v ∧ e2 ∈v ∧ e1 ≠ e2

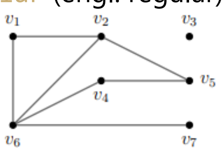


v_4 und v_5 sind benachbart, $v_4 \sim v_5$
e_1 und e_2 sind benachbart
e_2 und e_4 sind nicht benachbart
v_1 und v_5 sind nicht benachbart, $v_1 \not\sim v_5$

Exkurs Knotengrad

Der `Knotengrad` (engl. vertex degree) `deg(v)` ist die Anzahl der Kanten, auf denen v liegt.

- `Nachbarschaftsmenge` von v
  - N(v) = {u ∈ V : u ist benachbart zu v}.
  - Es gilt |N(v)| = deg(v)
- deg(v)/N(v) = | {u ∈ V | v ~ v} |
- `min. Knotengrad`: δ(G): = min v∈V deg(v)
- `max. Knotengrad`: Δ(G): = max v∈V deg(v)
- G heißt `r-regulär` (engl. regular), falls deg(v) = r für alle v ∈ V .



$d(v_1) = 2,\ d(v_2) = 3,\ d(v_3) = 0,\ d(v_4) = 2,\ d(v_5) = 2,\ d(v_6) = 4,\ d(v_7) = 1$
$N(v_1) = \{v_2, v_6\},\ N(v_2) = \{v_1, v_5, v_6\}$
$\delta(G) = 0$
$\Delta(G) = 4$

`Eingangsgrad` eines Knoten v ist die Anzahl der Kanten, für die v Endknoten ist
`Ausgangsgrad` eines Knoten v ist die Anzahl der Kanten für die v Startknoten ist.

8. Was besagt das Handshaking Lemma?

Die Anzahl der Knoten mit ungeradem Knotengrad ist gerade.

Beweis. Wir wissen von Satz 1.7:

$$\underbrace{2\,|E|}_{\text{gerade}} = \sum_{v \in V} \deg(v) = \underbrace{\sum_{\substack{v \in V \\ \deg(v)\ \text{gerade}}} \deg(v)}_{\text{gerade}} + \sum_{\substack{v \in V \\ \deg(v)\ \text{ungerade}}} \deg(v).$$

Daraus folgt:

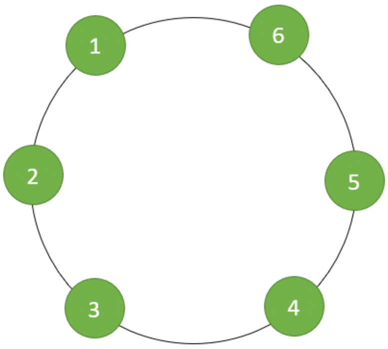
$$\sum_{\substack{v \in V \\ \deg(v)\ \text{ungerade}}} \deg(v)$$

```
ist gerade, woraus man direkt schließen kann, dass die Anzahl der Knoten
mit ungeradem Knotengrad, |{v ∈ V | deg(v) ungerade}|,
gerade ist.
```

9. Was ist ein regulärer Graph?

- Jeder Knoten hat gleich viele Kanten = jeder Knoten hat denselben Knotengrad = alle Knoten haben gleich viele Nachbarn

- deg(v) = r für alle v e V

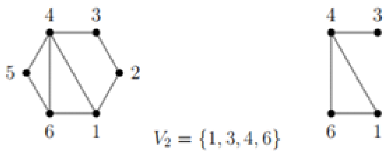
- Isolierter Knoten -> Grad/Nachbarschaft 0
- r-regulärer Graph: z.B. 2 regulärer Graph
- 

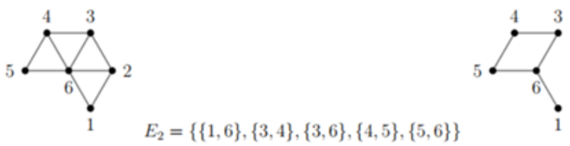
2-regulärer (ungerichteter) Graph

10. Ist Graph H ein Teilgraph von G (für zwei beliebige Beispiele von G und H)?

Ein Graph G2 = (V2, E2) heißt Teilgraph (Untergraph, engl. subgraph) eines Graphen G1 = (V1, E1), falls V2 ⊆ V1 und E2 ⊆ E1.

Knoten müssen gleiche Bezeichnungen haben
V2 darf nicht gerichtet sein, wenn V1 ungerichtet ist und umgekehrt!

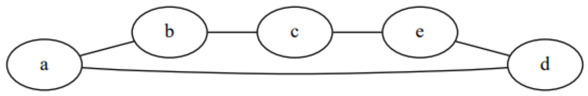
- `knoteninduzierter Teilgraph`
  - G = (V1, E1)
  - sei V2 ⊆ V1
  - Teilgraph : = (V2, [V2]^2 ∩ E1)
  - In Worten:
    - Man wählt eine Teilmenge V2 von V1 der Knoten aus und übernimmt alle Kanten von G zwischen Knoten aus V2
    - jeder Knoten in G2 ist gleich verbunden wie in G1
  - 

- `kanteninduzierter Teilgraph`
  - G = (V1, E1)
  - Teilmenge E2 von E1
  - Teilgraph : = (V2, E2)mit V2 = {v ∈ V1 : es gibt k ∈ E2 mit v ist Endknoten von k}
  - In Worten
    - Man wählt eine Teilmenge E2 von Kanten aus G aus und nimmt als Knotenmenge alle Endknoten von Kanten aus E2
  - 

- spannender Teilgraph:
  - G = (V, E)
  - Ein Teilgraph H = (V1, E1) heißt `spannender Teilgraph von G` falls V1 = V ist

`Beispiel:`

$G = (V,E), V = \{a,b,c,d,e\}, E = \{\{a,b\},\{a,d\},\{c,e\},\{b,c\},\{e,d\}\}$



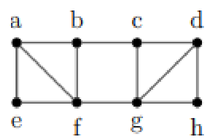
$H_1 = (\{a,b,d\},\{\{a,b\},\{a,d\}\}), H_2 = (\{a,b,c,d,e\},\{\{a,b\},\{a,d\},\{b,c\},\{d,e\}\}),$
$H_3 = (\{a,b,f\},\{\{a,b\}\}), H_4 = (\{d,e\},\emptyset)$

H1 = Teilgraph, da alle Knoten und Kanten von H1 in G enthalten sind!
H2 = Teilgraph, da alle Knoten und Kanten von H1 in G enthalten sind!
H3 = kein Teilgraph, da ein zusätzlicher Knoten f enthalten ist
H4 = Teilgraph, da die Leere Menge immer eine Teilmenge

11. Was ist eine Wanderung?

`Wanderung`

- Eine (u − v)-Wanderung W in G (engl.
  walk) ist eine *endliche Abfolge von Knoten und Kanten*
  - u -> Anfangsknoten, v -> Endknoten
  - Mächtigkeit |W| -> n, Länge der Wanderung (Kantenanzahl)
- Der Knoten u heißt Anfangsknoten von W, v heißt Endknoten von W. Die Länge |W| von W ist die Kantenanzahl n.
- `Knoten und Kanten können mehrfach vorkommen`
- es müssen aber nicht alle Knoten und Kanten besucht werden



- $W = (a, \{a,b\}, b, \{b,f\}, f, \{f,e\}, e, \{e,a\}, a, \{a,f\}, f, \{f,g\}, g, \{g,h\}, h)$
  $W = (a, b, f, e, a, f, g, h)$
- `geschlossene Wanderung` falls Anfangs- und Endpunkt übereinstimmen (u = v) sonst ist sie `offen`.
- `triviale Wanderung` W = v (v ∈ V)

`Weg` (engl. trail)

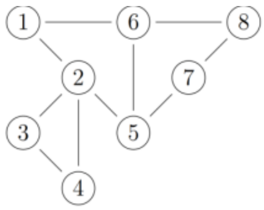
- falls alle Kanten in W verschieden sind.
- `geschlossener Weg` -> W = (u1, u2, . . . , un, u1), ui ∈ V heißt Zyklus oder Kreis (engl. cycle), falls ui ≠ uj, für alle 1 ≤ i ≠ j ≤ n.

`Pfad` (path)

- falls alle Knoten (und Kanten) in W verschieden sind.
- Daher kann es keine geschlossenen Pfade außer die Triviale Wanderung geben
- Sei G = (V, E) ein Graph und seien u, v ∈ V und sei weiters W eine (u−v)-Wanderung in G. Dann gibt es auch einen (u − v)-Pfad in G

`Pfad` -> `Weg` -> `Wanderung`
ein Pfad ist immer ein Weg; ein Weg muss aber nicht zwangsläufig ein Pfad sein!

11. Ist die Knotenfolge W (beliebiges Beispiel) eine Wanderung, Weg, Pfad oder Zyklus?



- $W_1 = (1,2,3,4,5,6,8), W_2 = (5,7,8), W_3 = (2,3,4,2,5), W_4 = (1,2,3,2,5,2,1),$
  $W_5 = (3,4,2,3), W_6 = \{1,2,3,4\}$

W1 = keines der drei (4 und 5 sind durch keine Kante verbunden)
W2 = Pfad (einmalige Knoten und Kanten)
W3 = Weg (doppelter Knoten, aber verschiedene Kanten)
W4 = geschlossene Wanderung (doppelte Knoten und Kanten)
W5 = geschlossener Weg (alle Knoten sind verschieden, und Anfang = Ende)
W6 ist in Mengenklammern, daher Menge an Knoten

12. Wann heißt ein Graph zusammenhängend?

G heißt zusammenhängend (engl. connected), falls es zu je
zwei Knoten aus V eine Wanderung von dem einen zum anderen gibt. Andernfalls ist G nicht zusammenhängend.

- zusammenhängend vs. nicht zusammenhängend
- 
- Die Menge der Wege bezeichnen wir mit W. Es gilt
  - $\mathcal{W} = \bigcup_{j=0}^{|V|} V^j.$
  - Auf der Menge an Wegen definieren wir eine Operation ◦, die zwei Wege aneinanderfügt.
  - (v1, . . . , vk) ◦ (w1, . . . , wj) := (v1, . . . , vk, w1, . . . , wj)

`leerer Weg`
also ein Weg/Pfad ohne einen Knoten, und schreiben hierfür ( ).

`Äquivalenzrelation`

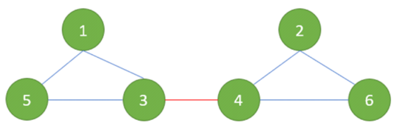
- G = (V, E)
- u, v ∈ V

- uRv ⇔ Es existiert eine Wanderung von u nach v.
- reflexiv: uRu da W = (u)
- symmetrisch: trivial, da man Wanderungen auch „von hinten lesen“ kann
- transitiv: uRv und vRw ⇒ uRw

13. Was ist in einem Graph eine Brücke? Geben Sie ein Beispiel in einem zusammenhängenden Graph mit |E| >= 5!

Eine Kante e ∈ E heißt Brücke (engl.bridge), wenn der Graph (V, E \ {e}) nicht zusammenhängend ist

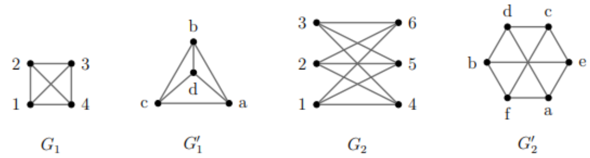
- Brücke ist die Kante die Graph verbindet, wenn diese entfernt wird dann zerfällt Graph in zwei Teilgraphen
- Wenn Graph ohne Kante e nicht zusammenhängend dann ist e eine Brücke
- `Zusammenhangskomponente` ist Anzahl der nicht zusammenhängenden Graphen Teile (Teilgraphen), wenn zusammenhängend dann 1 sonst n



14. Was bedeutet „zwei Graphen sind isomorph“?

- gleiche Anzahl an Kanten, Knoten
- gleiche Nachbarschaften (Knotengrade)
- Struktur ident
- |V1| = |V2| und |E1| = |E2| und deg(v_1) = deg(v_2) für alle Knoten in V1, V2
- naiv: die beiden Graphen sind bis auf die Beschriftung gleich

`isomorphe Graphen`



`Definition isomorph`

Zwei Graphen G1 = (V1, E1), G2 = (V2, E2) heißen isomorph (engl. isomorphic), falls es eine Abbildung f : V1 → V2 gibt, sodass

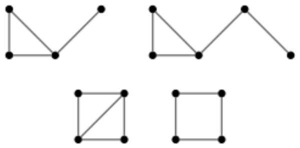
1. *f bijektiv*
2. *f „erhält Nachbarschaftsverhältnisse“, das heißt für alle $u, v \in V_1$ gilt $u \sim v \Leftrightarrow f(u) \sim f(v)$.*

Seien $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$ zwei isomorphe Graphen. Dann gelten:

- $|V_1| = |V_2|$
- $|E_1| = |E_2|$
- *für alle $v \in V_1$ gilt $\deg(f(v)) = \deg(v)$ für Isomorphismen $f\colon V_1 \to V_2$.*

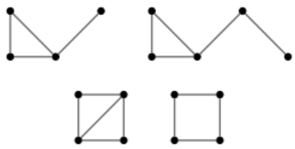
`nicht isomorphe Graphen`

Kanten bzw. Knoten sind unterschiedlich



`nicht isomorphe Graphen`

Nachbarschaftsverhältnis verletzt



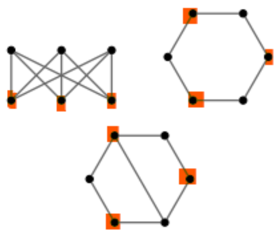
15. Was ist ein bipartiter Graph?

Man kann den Graphen in zwei Knotenmengen teilen, sodass es nur Kanten zwischen den beiden Knotenmengen gibt, aber nicht innerhalb

Sei $G = (V, E)$ ein Graph. G heißt bipartit (engl. bipartite), falls es eine Partition, $\{V_1, V_2\}$ mit $V = V_1 \cup V_2$, $V_1 \cap V_2 = \{\,\}$, der Knotenmenge V der folgenden Form gibt:

$$\text{für alle } u, v \in V_1 \Rightarrow u \not\sim v \text{ und für alle } u, v \in V_2 \Rightarrow u \not\sim v,$$

das heißt Kanten existieren höchstens zwischen Knoten aus V_1 und V_2.



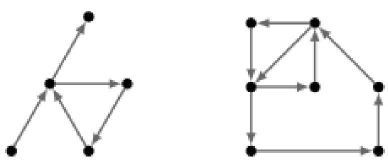
- Wie evaluieren?
  - Zähle Kanten Anzahl
  - `Breitensuche` (Ebene für Ebene durchsuchen)-> Starte bei Knoten und schaue alle Nachbarn an, dann die Nachbarn der Nachbarn (Stack)
  - `Tiefensuche` (Pro Strang bis zu den Blättern durchsuchen) -> wir gehen zuerst einen ganzen Pfad durch, alle Kindknoten aller Kindknoten

4. **Der vollständige bipartite Graph $K_{n,m}$**
$V = \{u_1, \ldots, u_n\} \cup \{v_1, \ldots v_m\}$,
$E = \{\{u_i, v_j\} : u_i \in \{u_1, \ldots, u_n\}, v_j \in \{v_1, \ldots v_m\}\}$.

16. Wann ist ein Graph eulersch?

Sei $G = (V, E)$ ein Graph. Ein Weg W heißt Eulerweg in G (engl. Euler trail), falls dieser Weg alle Kanten von G enthält.

Ein Eulerweg ist also eine Wanderung in G, welche jede Kante genau einmal benützt.



Ein Graph heißt eulersch oder Eulergraph (engl. eulerian), falls es in diesem Graphen einen geschlossenen Eulerweg gibt.
Man nennt einen geschlossenen Eulerweg oft Eulerkreis (engl. Euler tour), *selbst wenn es sich nicht um einen Kreis handelt, das heißt wenn Knoten mehrfach vorkommen*.

`Definition Eulersch`

Sei $G = (V, E)$ ein Graph und sei $W = (u_0, e_1, u_1, \ldots, e_n, u_n)$ ein Weg von u_0 nach u_n in G. Dann gelten:

1. *Falls W geschlossen ist ($u_0 = u_n$), dann ist für alle $0 \leq i \leq n$ die Anzahl der Kanten, die u_i als Endknoten haben, gerade, das heißt der Knotengrad von u_i ist gerade.*

Eulerscher Graph: wenn es einen Eulerkreis gibt -> gibt Eulerkreis immer dann, wenn der Grad aller Knoten gerade ist (jeder Knoten muss einen geraden Grad haben) + Graph muss zusammenhängend / geschlossen sein

2. *Falls W offen ist ($u_0 \neq u_n$), dann ist für alle $1 \leq i \leq n-1$ die Anzahl der Kanten von W, die u_i als Endknoten haben, gerade und die Anzahl der Kanten von W, auf denen u_0 beziehungsweise u_n liegen, ist ungerade.*

Sei G ein zusammenhängender Graph. Dann sind folgende Aussagen äquivalent.

1. *G ist eulersch.*
2. *Jeder Knoten von G hat einen geraden Grad.*

`Falls es genau zwei Knoten mit ungeradem Grad gibt, existiert ein offener Eulerweg; falls alle Knotengrade gerade sind, existiert ein geschlossener Eulerweg.`

• Eulersch (geschlossen) vs. Semi-eulersch (offen): das eine ist ein Zyklus und das andere nur ein Weg (Eulerweg -> zwei Knoten dürfen einen ungeraden Knotengrad haben: und zwar der Start und Endknoten)

- offener Eulerweg -> Haus des Nikolaus

• Eulerweg wenn Weg/Wanderung alle Kanten in G enthält, alle Kanten GENAU einmal
• Eulerkreis wenn ein geschlossener Eulerweg existiert

• Für einen Eulerkreis in einem vollständigen Graphen muss die Anzahl an Knoten ungerade sein, damit die Grade aller Knoten gerade sind
• Weg geschlossen, dann hat jeder Knoten eine gerade Anzahl an Degree
• Wenn offen dann genau 2 Knoten mit ungeradem Degree, sonst kein Eulergraph

17. Gegeben ein beliebiger Graph G, ist es hier möglich einen Eulerkreis zu konstruieren?

`Graph ist eulersch`

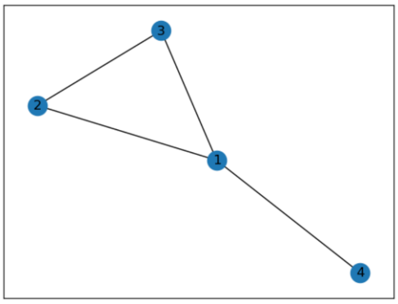
- Jeder Knoten von G hat einen geraden Grad == G ist eulersch

`Graph hat Eulerweg`

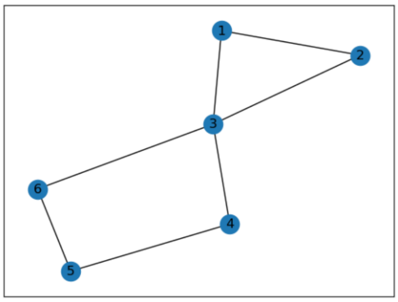
- Es gibt genau zwei oder keinen Weg mit ungeradem Grad. Alle anderen haben geraden Grad. == G hat einen Eulerweg
- Falls es genau zwei Knoten mit ungeradem Grad gibt, existiert ein offener Eulerweg; falls alle Knotengrade gerade sind, existiert ein geschlossener Eulerweg.

18. Geben Sie, falls möglich, den Eulerweg oder Eulerkreis in Graph G (beliebiger Graph) an!

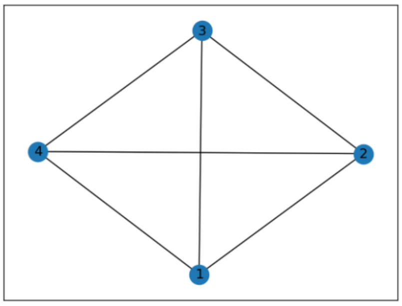
`kein Eulerkreis - kein Hamiltonkreis`



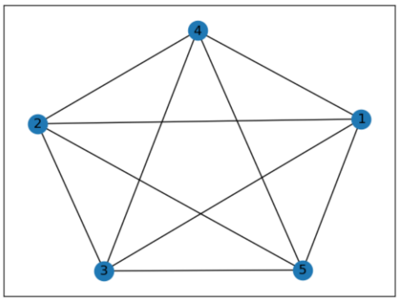
`Eulerkreis - kein Hamiltonkreis`



`kein Eulerkreis - Hamiltonkreis`



`Eulerkreis - Hamiltonkreis`



`gegebener Graph`

- Fleury durchspielen:
    - Zufälligen Startknoten wählen, von dort aus zufälliger Kante
    - Wenn Knoten mit ungeradem Grad existieren, eignen sich diese gut als Startknoten (siehe “Haus des Nikolaus”)
    - Kante nur wählen, wenn keine Brücke (außer beim letzten Knoten)
    - Mit Knoten am anderen Ende der Kante fortfahren

19. Welche Algorithmen kennen Sie zur Berechnung von Eulerkreisen? Was sind die Unterschiede?

- Zwiebelschalen-Algorithmus (wurde in VO nicht besprochen)

- Fleury Algorithmus

  - Findet Eulerkreis
  - muss zusammenhängender Graph sein
  - Initialisiere alle Kanten in Menge F, Lösungsmenge der Kanten ist leer
  - Wir beginnen bei Zufallsknoten und wählen Zufallskante
  - entferne Kante aus Menge F und gib zur Lösungsmenge hinzu (falls KEINE BRÜCKE)
  - neuer Knoten ist Ende von Kante
  - wiederhole Schritte

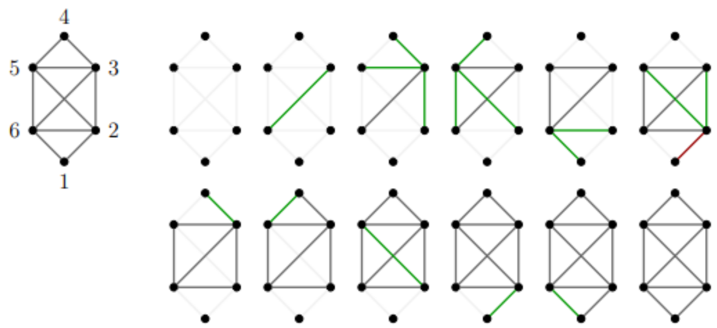


Abbildung 1.4: Eulerkreis im Haus des Nikolaus

Der Eulerkreis ist also $(6, 3, 5, 6, 2, 3, 4, 5, 2, 1, 6)$.

- Hierholzer Algorithmus (nicht im Skriptum)
  - identifiziert Zyklen und hängt diese zusammen
  - basiert auf Tiefensuche
  - alle Knoten weg dann Knotenliste
  - wird als zwei Stack aufgebaut, temporär und euler
  - wir nehmen Knoten und entfernen alle seine Kanten, dann hängen wir ihn dazu
  - wenn wir nicht alle Kanten aus dem Knoten löschen konnten, nimm einen Random Nachbarn und wiederhole die Schritte
  - nur Knoten ohne Nachbar kommt in Euler Stack
  - rekursiv, viel komplexer

20. Gegeben der Fleury Algorithmus in folgendem Zustand und nebenstehender Graph mit Kanten E \ F. Welche Kante würde als nächstes ausgewählt und warum?

- Auf Brücken achten!
- siehe Beschreibung zu Fleury 1 Frage vorher

21. Bei der Auswahl der nächsten Kante im Fleury Algorithmus muss eine mögliche Brücke immer zuletzt gewählt werden? Wieviele solcher Brücken kann es ausgehend von ein und demselben Knoten maximal geben (unter der Annahme, dass der Graph eulersch ist)?

Unter der Annahme, dass der Graph eulersch ist, kann es `maximal eine Brücke ausgehend von ein und demselben Knoten geben`. Ein eulerscher Graph ist ein Graph, der einen Eulerweg enthält, der jede Kante genau einmal durchläuft. In einem solchen Graphen können alle Knoten einen geraden Grad haben, außer möglicherweise zwei Knoten, die einen ungeraden Grad haen.

Angenommen, wir haben einen eulerschen Graphen und wählen einen Knoten als Ausgangspunkt für den Fleury-Algorithmus. Wenn dieser Knoten einen ungeraden Grad hat, wird die erste Kante, die wir wählen, zu einem anderen Knoten führen und den Grad des Ausgangsknotens auf gerade ändern. Da der Graph eulersch ist, können wir dann alle anderen Kanten durchlaufen und schließlich zu dieser ersten Kante zurückkehren, um den Eulerweg zu vervollständigen.

Wenn der Ausgangsknoten jedoch einen geraden Grad hat, können wir eine beliebige Kante als Startkante wählen, da es keine Brücken gibt, die vermieden werden müssen. Wir können den gesamten Eulerweg durchlaufen und schließlich zum Startknoten zurückkehren, ohne eine Brücke zu überqueren.

`Im Eulergraph selbst darf keine Brücke existieren` (Ansonsten kann kein Eulerkreis enthalten sein, denn dann wäre es nur ein Eulerpfad und somit auch kein Eulergraph mehr)

- In einem normalen Graphen kann es so viele Brücken geben, wie es Kanten gibt

22. Was bedeutet es, wenn ein Graph die Hamilton Eigenschaft besitzt?

2. Diskutieren Sie die Euler- bzw. Hamiltoneigenschaft für vollständige Graphen K_n mit n > 3!
3. Diskutieren Sie die Lösbarkeit des Eulerkreisproblems mit dem des Hamiltonkreisproblems!
4. Skizzieren Sie einen Algorithmus zur Lösung des Hamiltonkreisproblems!
5. Welche Algorithmen für die kürzeste Wegfindung kennen Sie? Vergleichen Sie diese und diskutieren Sie die Anwendbarkeit dieser Algorithmen!
6. Gegeben der Dijkstra Algorithmus in folgendem Zustand. Welche Kante würde als nächstes ausgewählt? Wie lautet der kürzeste Weg? Welchen Weg findet der Dijkstra Algorithmus?
7. Wie können Sie den Dijkstra Algorithmus potenziell noch beschleunigen?
8. Wie lautet die Abbruchbedingung der bidirektionalen Suche?
9. Welche Eigenschaft muss die Heuristik für den A* erfüllen?
10. Gegeben der A* Algorithmus in folgendem Zustand. Welche Kante würde als nächstes ausgewählt? Wie lautet der kürzeste Weg? Welchen Weg findet der A* Algorithmus?
11. Was ist ein Spannbaum, was ist ein minimaler Spannbaum?
12. Ist der Spannbaum aus dem Dijkstra Algorithmus minimal?
13. Welche Algorithmen für die Berechnung von Spannbäumen kennen Sie? Diskutieren Sie die Unterschiede!
14. Ist der minimale Spannbaum eindeutig, wann / wann nicht?

15. Genügt es einen minimalen Spannbaum zu finden um in einem Graph einfach alle kürzesten Wege zu berechnen? Warum bzw. warum nicht?
16. Was ist ein Webgraph?
17. Erklären Sie das Random Surfer Modell, welche Annahmen sind in dem Modell enthalten?
18. Was bedeutet der Dämpfungsfaktor im Random Surfer Modell?
19. Der Page Rank ist eine spezielle Form der Eigenvektorzentralität, was beschreibt der Page Rank?
20. Wie kann der Page Rank für die Suche verwendet werden?
21. Welches Verfahren wird für die Berechnung des Page Rank verwendet?
22. Gegeben ein beliebiger Graph G, schätzen Sie die Reihenfolge der Knoten die sich durch den Page Rank ergeben würde! Begründen Sie Ihre Schätzung!
23. Beschreiben Sie 3 Zentralitätsmaße für Graphen!
24. Was ist ein Netzwerk und was ist ein Fluss?
25. Geben Sie eine Definition des maximalen Flussproblems!
26. Zeichnen Sie die Residualgraphen nach jeder Iterationen des Ford-Fulkerson-Algorithmus für ein beliebiges Netzwerk N für die ersten 3 Iterationen!
27. Wie lässt sich ein augmentierender Pfad finden?
28. Wann resultiert der Ford-Fulkerson-Algorithmus in einer ganzzahligen Lösung für maxflow?
29. Was ist ein Schnitt und wie hängen maximaler Fluss und minimaler Schnitt zusammen?
30. Wie lautet die Standardform der linearen Programmierung?
31. Aus welchen 5 Teilen ist ein lineares Programm aufgebaut?
32. Welches Problem ergibt sich wenn in einem linearen Programm diskrete Entscheidungsvariablen definierent werden?
33. Wie werden Ungleichungen bzw. Gleichungen in einem linearen Programm abgebildet?
34. Was bedeutet es wenn künstliche Variablen in der optimalen Lösung eines linearen Programms einen Wert ungleich 0 annehmen?
35. Eine chinesische Postbotin muss in einem Dorf Information zu einer bevorstehenden Wahl austragen die bereits in wenigen Tagen stattfinden soll. In jeder Straße des Dorfes stehen Häuser die sie besuchen muss. Sie ist besorgt, dass möglicherweise nicht alle die Information rechtzeitig erhalten. Wie kann man das Austragen der Information effizient gestalten? Welches Problem verbirgt sich dahinter bzw. wie könnte man es erweitern?
36. Eine Stadt möchte die Wasserversorgung in einem bestimmten Gebiet verbessern. Bewohner beschweren sich, dass an heißen Tagen zuwenig Wasser aus der Leitung kommt.

Die Stadt stellt jedoch fest, dass genügend Wasser im Speicher vorhanden ist. Wie könnten Sie die Stadt bzw. die Bewohner hier unterstützen?

37. Das Management eines Einkaufszentrums stellt fest, dass trotz attraktiver Geschäfte in einem Bereich nur vergleichsweise wenige Kunden zu finden sind. Das Management möchte herausfinden warum und wie das zu verbessern wäre. Wie könnten Sie das Management unterstützen?

38. Ein Essenslieferant beobachtet anhand des Kundenfeedbacks nach einer Lieferung, dass bestimmte Auslieferer besser mit bestimmten Kunden zurechtkommen als mit anderen. Der Lieferant möchte die Kundenzufriedenheit steigern. Wie könnten Sie den Lieferant hier unterstützen?

39. Die Bestimmung von Lehrveranstaltungszeiten, sowie die Raumbelegung sind an einer Universität für die administrativen Kräfte große Herausforderungen und nehmen jedes Semester etliche Stunden in Anspruch. Versuchen Sie das Problem so weit als möglich als lineares Programm zu skizzieren. Beschreiben Sie Mengen, Parameter, Entscheidungsvariablen, Ziele und mögliche Nebenbedingungen, möglichst in mathematischer Form. Ein paar Anhaltspunkte:
    (1) ein Raum kann zu einem Zeitpunkt nur eine LVA abhalten,
    (2) Personen können sich zur selben Zeit in nur einem Raum befinden,
    (3) der Raum muss genügend Kapazität haben damit alle Platz finden,
    (4) der Raum muss die Anforderungen an die LVA berücksichtigen (PC Plätze, Beamer, Tafel, Whiteboard,
    onlinefähig). Was könnten mögliche Ziele für so ein Modell sein?