Graphentheorie

Graph: $G = (V, E)$

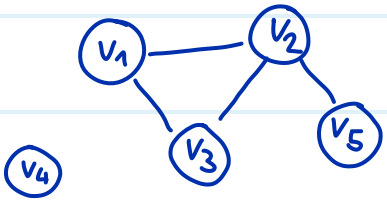
Knotenmenge $V = \{v_1, v_2, \ldots, v_n\}$ meistens ist $|V| = n$

Kantenmenge $E \subseteq \{\{u, v\} \mid u, v \in V\}$ meistens ist $|E| = m$

benachbart/adjazent: $u, v \in V$ sind benachbart, falls es eine Kante zwischen u und v gibt, d.h. $\{u, v\} \in E$

anliegend/inzident: Eine Kante $e \in E$ ist inzident zu Knoten $v \in V$, falls v einer der beiden Endpunkte von e ist

- For $v \in V$, the **degree** $\deg(v)$ of v (german "Knotengrad") is the number of edges that are incident to v.



$V = \{a, b, c, d, e\}$

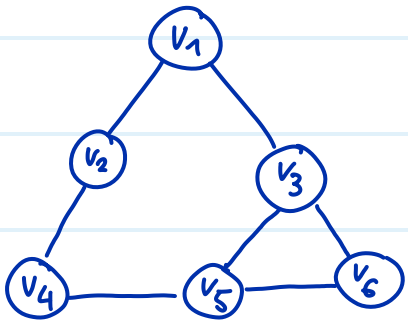
$E = \{\{a, b\}, \{b, c\}, \{c, e\}, \{e, b\}\}$

- A sequence of vertices $(v_0, v_1, \ldots, v_k)$ (with $v_i \in V$ for all i) is a **walk** (german "Weg") if $\{v_i, v_{i+1}\}$ is an edge for each $0 \leq i \leq k-1$. We say that v_0 and v_k are the **endpoints** (german "Startknoten" and "Endknoten") of the walk. The **length** of the walk $(v_0, v_1, \ldots, v_k)$ is k.
- A sequence of vertices $(v_0, v_1, \ldots, v_k)$ is a **closed walk** (german "Zyklus") if it is a walk, $k \geq 2$ and $v_0 = v_k$.

Achtung: Zyklus ≠ cycle

- A sequence of vertices $(v_0, v_1, \ldots, v_k)$ is a **path** (german "Pfad") if it is a walk and all vertices are distinct (i.e., $v_i \neq v_j$ for $0 \leq i < j \leq k$).
- A sequence of vertices $(v_0, v_1, \ldots, v_k)$ is a **cycle** (german "Kreis") if it is a closed walk, $k \geq 3$ and all vertices (except v_0 and v_k) are distinct.

How many different paths from v_1 to v_6 are there and what length do they have?



How many different walks with endpoints v_1 and v_6 are there?

How many different cycles are there?

What are all possible lengths for a walk with endpoints v_1 and v_6?

What are all possible lengths for a path with endpoints v_1 and v_6?

- An **Eulerian walk** (german "Eulerweg") is a walk that contains every edge exactly once.
- A **closed Eulerian walk** (german "Eulerzyklus") is a closed walk that contains every edge exactly once.

} Algorithmus in $O(n+m)$

- A **Hamiltonian path** (german "Hamiltonpfad") is a path that contains every vertex.
- A **Hamiltonian cycle** (german "Hamiltonkreis") is a cycle that contains every vertex.

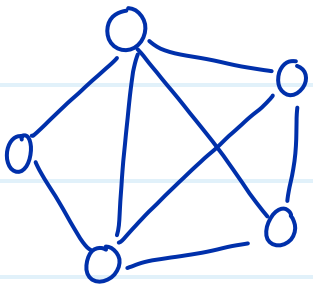
} NP-Vollständig

Sei G zusammenhängend

G hat Eulerweg <=> Alle Knotengrade ausser höchstens zwei sind gerade

(ungerade Knotengrade sind Start- und Endpunkte)

G hat Eulerzyklus <=> Alle Knotengrade sind gerade

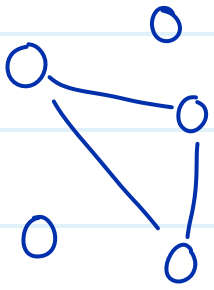


Eulerweg?

Eulerzyklus?

Hamiltonpfad?

Hamiltonkreis?



Eulerweg?

Eulerzyklus?

Hamiltonpfad?

Hamiltonkreis?

Sei G beliebig.

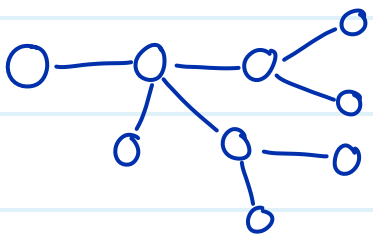
G hat Eulerweg <=> Alle Knotengrade ausser höchstens zwei sind gerade und alle Kanten sind in derselben ZHK

G hat Eulerzyklus <=> Alle Knotengrade sind gerade und alle Kanten sind in derselben ZHK

- For $u, v \in V$, we say u **reaches** v (or v is **reachable** from u; german "u erreicht v") if there exists a walk with endpoints u and v, or equivalently, there exists a path with endpoints u and v.
- A **connected component** of G (german "Zusammenhangskomponente") is an equivalence class of the (equivalence) relation defined as follows: Two vertices $u, v \in V$ are equivalent if u reaches v.
- A graph G is **connected** (german "zusammenhängend") if for every two vertices $u, v \in V$, u reaches v, or equivalently, if there is only one connected component.

- A graph G is a **tree** (german "Baum") if it is connected and has no cycles.

$m = n - 1$



Handshake Lemma: $\sum_{v \in V} \deg(v) = 2|E|$

Beweis: Jede Kante ist zu genau zwei Knoten inzident.

(Wenn wir eine Kante hinzufügen, erhöht sich der Grad von genau zwei Knoten, nämlich den beiden Endpunkten der Kante um 1)

Aussage Jeder Graph G, in dem alle Knoten einen Grad von genau 2 haben, ist zusammenhängend.

Aussage Sei G ein Graph mit $n \geq 3$ Knoten. Wenn G zusammenhängend ist und $m = n - 1$ Kanten hat, dann muss G einen Knoten vom Grad 1 haben.

Aussage Wenn es in einem Graphen einen Weg (walk) von Knoten a nach b gibt und einen Weg von b nach c, dann gibt es immer auch einen Pfad (path) von a nach c.

Aussage Wenn ein Graph G einen Hamiltonpfad enthält, dann muss G zusammenhängend sein.

Aussage Wenn ein Graph G einen Eulerweg (aber keinen Eulerzyklus) enthält, dann hat er genau zwei Knoten mit ungeradem Knotengrad.

Aussage Jeder Graph G mit $n \geq 3$ Knoten, in dem jeder Knoten einen Grad von mindestens $n/2$ hat, muss zusammenhängend sein.

Eulerzyklus Algorithmus in O(E)

1. **Start:** Achilles startet am Startknoten s und findet einen beliebigen ersten Zyklus C.
2. **Initialisierung:** Die Schildkröte setzt ihre Haupttour T gleich diesem Zyklus ($T = C$).
3. **Wanderung:** Die Schildkröte beginnt, ihre Tour T ab s abzulaufen.
4. **Prüfung:** An *jedem* Knoten v, den sie auf ihrer Tour T erreicht, prüft die Schildkröte, ob von v noch **unbesuchte** Kanten ausgehen.
   - **Fall A (Keine Abzweigung):** Wenn nein, läuft sie einfach weiter zum nächsten Knoten gemäß T.
   - **Fall B (Abzweigung):** Wenn ja, **stoppt** die Schildkröte bei v.
5. **Neuer Zyklus:** Achilles wird bei v losgeschickt. Er läuft *ausschließlich* über unbesuchte Kanten, bis er einen neuen Zyklus C_{neu} gebildet hat und exakt zu v zurückkehrt.
6. **Einfügen:** Die Schildkröte fügt den neuen Zyklus C_{neu} an der Position v in ihre Haupttour T ein.
   - *(Beispiel: T war `A -> v -> B`. C_{neu} ist `v -> X -> v`. Neues T ist `A -> [v -> X -> v] -> B`.)*
7. **Fortsetzung:** Die Schildkröte setzt ihre Wanderung *sofort* auf dem neu eingefügten Pfad C_{neu} fort. Wenn sie C_{neu} beendet hat, folgt sie dem Rest der ursprünglichen Tour T (im Beispiel: ...nach B).
8. **Ende:** Der Algorithmus ist beendet, sobald die Schildkröte ihre *gesamte* (jetzt erweiterte) Tour T einmal durchlaufen hat, ohne Achilles (in Schritt 4B) erneut losschicken zu müssen.