Wiederholung

- Graph: $G = (V, E)$ V endl. Menge Knoten von G $E \subseteq \{\{u,v\} \mid u \neq v \in V\}$ Kanten von G

  - $n_G = |V|$, $m_G = |E|$

  - $\{u,v\} \in E$: $uv = vu = \{u,v\} =: e$ u,v <u>Endknoten</u> von e

    u,v adjazent = benachbart

    u,v inzident zu e

  - $\Gamma(v)$ = Menge der Nachbarn von v $(v \notin \Gamma(v))$

    $\deg(v) := |\Gamma(v)|$

  - $\sum_{v \in V} \deg(v) = 2 \cdot m_G$ Handschlagslemma

- Adjazenzmatrix von $G=(V,E)$, $V=\{1,2,\dots,n\}$

  $(a_{ij}) \in \{0,1\}^{n\times n}$, $a_{ij} = \begin{cases} 1, & \text{falls } ij \in E \\ 0, & \text{sonst} \end{cases}$

- Adjazenzliste: $\left(\Gamma(1), \Gamma(2), \dots, \Gamma(n)\right)$

- Inzidenzmatrix: $(b_{ij}) \in \{0,1\}^{n\times m}$

- Teilgraphen $G=(V,E)$, $G'=(V',E')$
  - $G' \leq G$ falls $V' \subseteq V$, $E' \subseteq E$
  - G' induzierter Teilgraph auf V', falls $V' \subseteq V$ und $E' = \{uv \mid u,v \in V', uv \in E\}$

- <u>Kantenzug</u>: $(v_0, v_1, \dots, v_\ell)$, $v_i \in V$, $v_{i-1}v_i \in E$ $\forall i$ (v_0-v_ℓ-Kantenzug)

  <u>Länge</u> ℓ; <u>geschlossen</u>, falls $v_\ell = v_0$; Länge 0: (v_0)

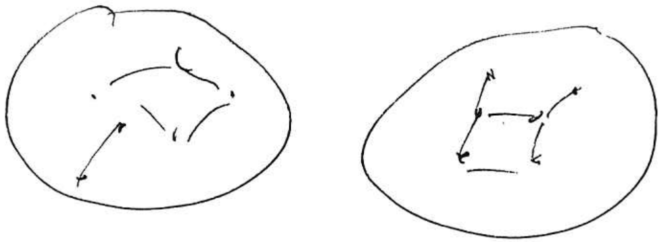
  <u>Pfad</u>: falls $v_0, v_1, \dots, v_\ell$ paarw. versch.

  <u>Kreis</u>: falls $\ell \geq 3$, und geschl. u. $(v_0, \dots, v_{\ell-1})$ Pfad

  <u>Tour</u>: geschlossen u. $v_{i-1}v_i$ paarw. versch. $(i = 1, \dots, \ell)$

- $u \sim v$ (verbunden) $:\Longleftrightarrow$ $\exists$ u-v-Kantenzug

  ÄR auf V; induzierten Teilgraphen auf ÄK: Z.K.

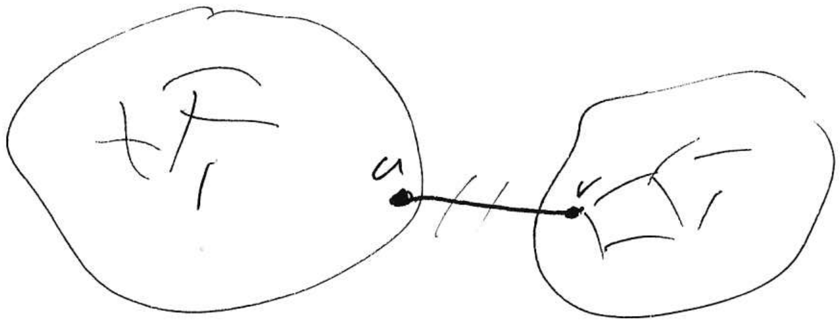


r_G: Anzahl der Z.K.

- $u, v \in V$, $u \neq v$, $uv \in E$

  $uv \in E$ Brücke $:\Longleftrightarrow$ $r_{(V, E \setminus \{uv\})} > r_G$

  [$\Longleftrightarrow$ $\nexists$ u-v-Kantenweg in G, der nicht über uv führt]



Brücken (Forts.)

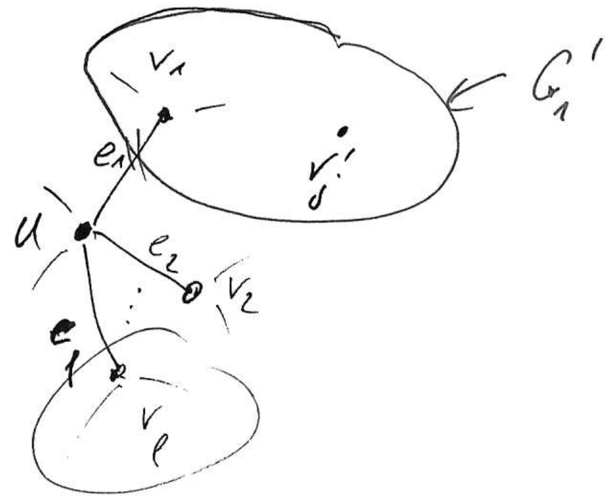
Es sei $G = (V, E)$ ein Graph und $l \in \mathbb{N}$.

Satz
Ist $u \in V$ zu l Brücken inzident, so besitzt G mindestens l von u verschiedene Knoten von ungeradem Grad.

Folgerung

Haben in einem Graphen alle Knoten geraden Grad, so besitzt er keine Brücken.

Beweis des Satzes



ℓ Brücken $e_1, \dots, e_\ell$

$G \setminus \{e_1\}$ hat Brücken $e_2, \dots, e_\ell$

v_1 liegt in einer anderen Z.K. von $G \setminus \{e_1\}$ als u

$\Rightarrow$ In $G' := (V, E \setminus \{e_1, \dots, e_\ell\})$ liegen $v_1, \dots, v_\ell$ in paarw. versch. Z.K. $G_1', G_2', \dots, G_\ell'$

$\deg_G(v_j)$ gerade $\Rightarrow$ $\deg_{G'}(v_j)$ ungerade

$\deg_G(v_j)$ ungerade ~~[$\Rightarrow$ $\deg_{G'}(v_j)$ gerade]~~ gut

$\deg_{G'}(v_j)$ ungerade $\Rightarrow$ $\exists\, v_j' \in G_j'$ mit $\deg_{G'}(v_j')$ ungerade, $v_j \neq v_j'$

$\Rightarrow$ $\deg_G(v_j') = \deg_{G'}(v_j')$ ungerade gut.

Distanz

Es sei $G = (V, E)$ ein Graph.

Definition

Es seien $v, w \in V$.

- Ist $v \sim w$, dann sei

$$d(v, w) := \min\{l \in \mathbb{N}_0 \mid \text{in } G \text{ ex. } v\text{-}w\text{-Pfad der Länge } l\} \in \mathbb{N}_0.$$

- Ist $v, w \in V$ mit $v \not\sim w$, dann sei $d(v, w) := \infty$.
- Wir nennen $d(v, w)$ die *Distanz* zwischen v und w.

Distanz

Es sei $G = (V, E)$ ein Graph.

Bemerkung

Für alle $v, w \in V$ gelten:

- $d(v, w) = 0 \Leftrightarrow v = w$,
- $d(v, w) < \infty \Leftrightarrow v \sim w$.

G ist genau dann zusammenhängend, wenn gilt:
$d(v, w) < \infty$ für alle $v, w \in V$.

Breitensuche

Es sei $G = (V, E)$ ein Graph und $w \in V$.

Die *Breitensuche* ist ein Algorithmus, der, beginnend mit $w \in V$, alle Knoten der Zusammenhangskomponente von w mit aufsteigender Distanz zu w durchläuft.

Anwendungen

- Berechnung der Zusammenhangskomponenten von G.
- Berechnung der Distanzen $d(v, w)$ für v in der Zusammenhangskomponente von w.
- Berechnung kürzester Pfade von jedem v zu w.

Breitensuche (Forts.)

```
BREITENSUCHE(Γ, w)
initialisiere array d[1, ..., n] mit allen Einträgen gleich ∞
initialisiere array p[1, ..., n] mit allen Einträgen gleich NIL
initialisiere leere queue Q (FIFO)
d[w] ← 0
INSERT(Q, w)
while Q ist nicht leer
do v ← EXTRACT(Q)
    for u ∈ Γ(v)
    do if d[u] = ∞
          then INSERT(Q, u)
               d[u] ← d[v] + 1
               p[u] ← v
return d, p
```

Breitensuche (Forts.)

Kommentare zum Algorithmus)

- Eingabe:
  - Γ: Adjazenzliste des Graphen $G = (V, E)$ mit $V = \underline{n}$
  - w: Knoten $w \in V$
- Der `array` $d[1, \ldots, n]$ enthält nach der Terminierung an Position v den Wert $d(w, v)$.
- Der `array` $p[1, \ldots, n]$ enthält nach der Terminierung an Position v einen Knoten u, der auf einem w-v-Pfad der Länge $d(w, v)$ unmittelbar vor v kommt.
- `queue` ist eine Warteschlange im „First-in-first-out“-Modus
- Der Aufruf $\textsc{Insert}(Q, x)$ hängt das Element x an das Ende der Warteschlange.
- Der Aufruf $\textsc{Extract}(Q)$ entnimmt das Element, das am Anfang der Warteschlange steht.

Breitensuche (Forts.)

Bemerkung

Der Verlauf der Breitensuche und das Ergebnis p hängen von der Anordnung der Mengen $\Gamma(v)$ in der Adjazenzliste von G ab.

Beispiel

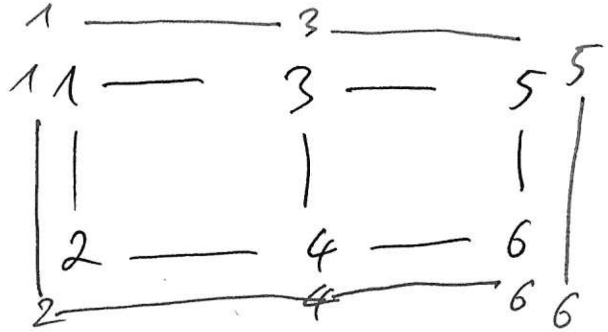
$$\begin{tikzcd}
1 \arrow[r, dash] \arrow[d, dash] & 3 \arrow[r, dash] \arrow[d, dash] & 5 \arrow[d, dash] \\
2 \arrow[r, dash] & 4 \arrow[r, dash] & 6
\end{tikzcd}$$

$\Gamma(1) = [2,3]$

$\Gamma(2) = [1,4]$

$\Gamma(3) =$

Breitensuche (Forts.)



Beispiel

Die Listen $\Gamma(v)$ sind aufsteigend angeordnet.

d	p	Q	v	$\Gamma(v)$	$d[u] = \infty$
$[0, \infty, \infty, \infty, \infty, \infty]$	$[-, -, -, -, -, -]$	$[1]$	1	$[2, 3]$	$[2, 3]$
$[0, 1, 1, \infty, \infty, \infty]$	$[-, 1, 1, -, -, -]$	$[2, 3]$	2	$[1, 4]$	$[4]$
$[0, 1, 1, 2, \infty, \infty]$	$[-, 1, 1, 2, -, -]$	$[3, 4]$	3	$[1, 4, 5]$	$[5]$
$[0, 1, 1, 2, 2, \infty]$	$[-, 1, 1, 2, 3, -]$	$[4, 5]$	4	$[2, 3, 6]$	$[6]$
$[0, 1, 1, 2, 2, 3]$	$[-, 1, 1, 2, 3, 4]$	$[5, 6]$	5	$[3, 6]$	$[]$
$[0, 1, 1, 2, 2, 3]$	$[-, 1, 1, 2, 3, 4]$	$[6]$	6	$[4, 5]$	$[]$
$[0, 1, 1, 2, 2, 3]$	$[-, 1, 1, 2, 3, 4]$	$[]$			

Breitensuche (Forts.)

Beispiel

Die Listen $\Gamma(v)$ sind absteigend angeordnet.

d	p	Q	v	$\Gamma(v)$	$d[u] = \infty$
$[0, \infty, \infty, \infty, \infty, \infty]$	$[-, -, -, -, -, -]$	$[1]$	1	$[3, 2]$	$[3, 2]$
$[0, 1, 1, \infty, \infty, \infty]$	$[-, 1, 1, -, -, -]$	$[3, 2]$	3	$[5, 4, 1]$	$[5, 4]$
$[0, 1, 1, 2, 2, \infty]$	$[-, 1, 1, 3, 3, -]$	$[2, 5, 4]$	2	$[4, 1]$	$[]$
$[0, 1, 1, 2, 2, \infty]$	$[-, 1, 1, 3, 3, -]$	$[5, 4]$	5	$[6, 3]$	$[6]$
$[0, 1, 1, 2, 2, 3]$	$[-, 1, 1, 3, 3, 5]$	$[4, 6]$	4	$[6, 3, 2]$	$[]$
$[0, 1, 1, 2, 2, 3]$	$[-, 1, 1, 3, 3, 5]$	$[6]$	6	$[5, 4]$	$[]$
$[0, 1, 1, 2, 2, 3]$	$[-, 1, 1, 3, 3, 5]$	$[]$			

Tiefensuche

Bemerkung

- Die *Tiefensuche* ist ein Algorithmus mit der gleichen Ein- und Ausgabe wie die Breitensuche.
- In jedem Schritt der Breitensuche wird die Distanz zu w möglichst beibehalten.
- In jedem Schritt der Tiefensuche wird die Distanz zu w möglichst vergrößert.
- Sie wird realisiert, indem die queue (FIFO) durch einen stack (LIFO= „Last-in-first-out“) ersetzt wird.

Hamiltonkreise und Eulertouren

Definition

Es sei $G = (V, E)$ ein Graph.

- Ein Kreis der Länge n_G in G heißt *Hamiltonkreis*.
- Eine Tour der Länge m_G in G heißt *Eulertour*.

Hamiltonkreise und Eulertouren (Forts.)

Es sei $G = (V, E)$ ein Graph.

Bemerkung

- Ein geschlossener Kantenzug $(v_0, \ldots, v_l)$ ist genau dann ein Hamiltonkreis, wenn in der Auflistung $v_0, \ldots, v_{l-1}$ jeder Knoten aus V genau einmal vorkommt.

- Ein geschlossener Kantenzug $(v_0, \ldots, v_l)$ ist genau dann eine Eulertour, wenn in der Auflistung $v_0 v_1, v_1 v_2, \ldots, v_{l-1} v_l$ jede Kante aus E genau einmal vorkommt.

Definition

Ein (nicht notwendig geschlossener) Kantenzug $(v_0, \ldots, v_l)$ heißt *Eulerzug*, wenn in der Auflistung $v_0 v_1, v_1 v_2, \ldots, v_{l-1} v_l$ jede Kante aus E genau einmal vorkommt.

Eulertouren

Es sei $G = (V, E)$ ein Graph.

Bemerkung

Jeden Knoten v, den ich auf einer Kante der Tour erreiche, muss ich auf einer anderen Kante verlassen.

- Existiert in G eine Eulertour, so gelten:
  - Alle Knoten von G haben geraden Grad, und
  - höchstens eine Zusammenhangskomponente von G ist nicht-trivial.

- Existiert in G ein Hamiltonkreis, so gelten:
  - Jeder Knoten von G hat Grad ≥ 2, und
  - G ist zusammenhängend und $n_G \geq 3$.

Hamiltonkreise und Eulertouren (Forts.)

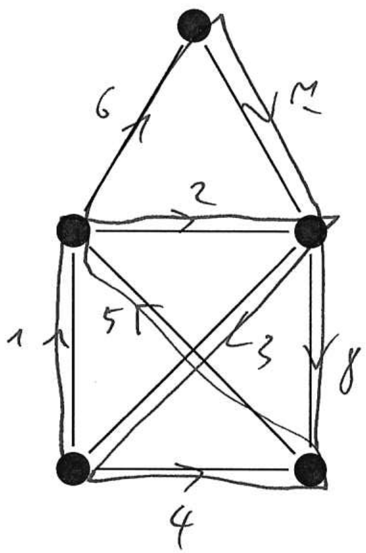
Beispiel

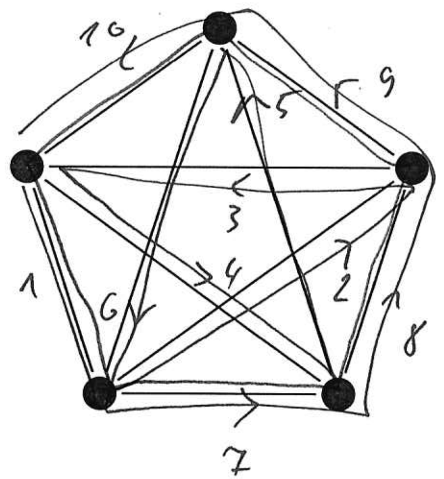
- Das Straßennetz einer Stadt sei durch einen Graphen modelliert (Knoten: Kreuzungen, Kanten: Straßenabschnitte).

  Der Fahrer eines Schneeräumfahrzeuges sucht eine Eulertour.

- Der Graph „Haus vom Nikolaus“ besitzt einen Eulerzug, aber keine Eulertour.

-





Eulertouren (Forts.)

Es sei $G = (V, E)$ ein zusammenhängender Graph.

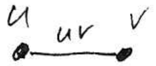
Satz

Hat G genau zwei Knoten u, v mit ungeradem Grad, dann existiert ein u-v-Eulerzug in G.

Folgerung

Der Graph G besitzt genau dann eine Eulertour, wenn alle Knoten von G geraden Grad haben.

Beweis des Satzes: Induktion nach m_G

$m_G = 1$: 

$m_G > 1$: Sei o.B.d.A. $\deg(u) \geq \deg(v)$

1. **Fall**: $\deg(u) > 1$

$\Rightarrow \deg(u) \geq 3$

u ist höchstens zu einer Brücke inzident

$\Rightarrow \exists\, w \in V$ adjazent zu u mit $w \neq v$ und w keine Brücke

$\Rightarrow G' := (V, E \setminus \{uw\})$ ist zusammenhängend,

w, v sind die einzigen Knoten von G' von ungeradem Grad

I.V. $\Rightarrow \exists$ w-v Kantenzug in G'.

2. **Fall**: $\deg(u) = 1$. $\Rightarrow \deg(v) \leq \deg(u) = 1$, d.h. $\deg(v) = 1$

$\Rightarrow \exists$ genau eine Kante $uw \in E$ und $w \neq v$, da $m_G > 2$

Setze $G' := (V \setminus \{u\}, E \setminus \{uw\})$. Weiter wie oben. $\blacksquare$

Eulertouren (Forts.)

Es sei $G = (V, E)$ ein zusammenhängender Graph, dessen Knoten geraden Grad haben. Die folgende Prozedur FLEURY berechnet eine Eulertour.

```
FLEURY(V, E)
initialisiere leere Liste T
v ← beliebiger Knoten aus V
APPEND(T, v)
while E ist nicht leer
do if deg v = 1
        then w ← einziger Nachbar von v
        else  w ← ein Nachbar von v mit vw keine Brücke
    APPEND(T, w)
    E ← E \ {vw}
    v ← w
return T
```

Beweis der Korrektheit von Fleury:

1. Durchlauf der while-Schleife: $T = [u, w]$

   $G' := (V, E \setminus \{vw\})$ zus. hgd., hat genau zwei ungerade Knoten

2. Andere Durchläufe der while-Schleife: v_0 Startpunkt

   v_0, v haben ungeraden Grad

   (a) $\deg(v) = 1$ $\qquad$ $vw \in E$

   (a1) $w = v_0$ $\qquad$ fertig

   (a2) $w \neq v_0$ $\qquad$ $\Rightarrow$ w geraden Grad

   (b) $\deg(v) > 1$ $\qquad$ Nicht-Brücke vw ex., da $\deg(v) \geq 2$, $w \neq v_0$

In (a) $\quad G' := (V' \setminus \{v\}, E' \setminus \{vw\})$ $\qquad$ $\deg_{G'}(w)$ ungerade

In (b) $\quad G' := (V', E' \setminus \{vw\})$ $\qquad$ — " —

Hamiltonkreise

Satz
Es sei $G = (V, E)$ ein zusammenhängender Graph mit $n_G \geq 3$.

Falls für alle $u, v \in V$ mit $u \neq v$ und $uv \notin E$ gilt

$$\deg u + \deg v \geq n_G,$$

so besitzt G einen Hamiltonkreis.

Bemerkung
Erfüllt G die Voraussetzungen des Satzes und ist n_G gerade, so gilt

$$m_G \geq \frac{n_G^2}{4}.$$

<u>Beweis des Satzes</u> $G = (V, E)$, K_n vollständige Graph auf V

$(v_1, \dots, v_n)$ Permutation von V

$(v_1, \dots, v_n, v_1)$ Kreis der Länge n in K_n

Sei $r :=$ Anzahl der Kanten von $(v_1, v_2, \dots, v_n, v_1)$ in E

<u>$r = n$</u> OK

<u>$r < n$:</u> O.B.d.A. $v_1 v_2 \notin E$

<u>Beh</u>: $\exists\, i \in \{3, 4, \dots, n\}$ mit $\underline{v_1 v_{i-1}}, v_2 v_i \in E$

Ist die Beh. bewiesen, dann ist

$(v_1, v_{i-1}, v_{i-2}, \dots, v_2, v_i, v_{i+1}, \dots, v_n, v_1)$ Kreis der Länge n in K_n

mit $\geq r+1$ Kanten in E

(da $v_1 v_2$, $v_{i-1} v_i$ ersetzt werden durch $v_1 v_{i-1}$, $v_2 v_i$).