

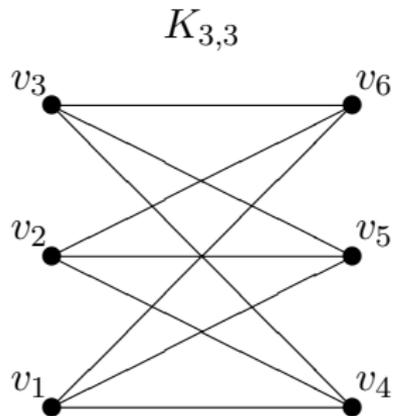
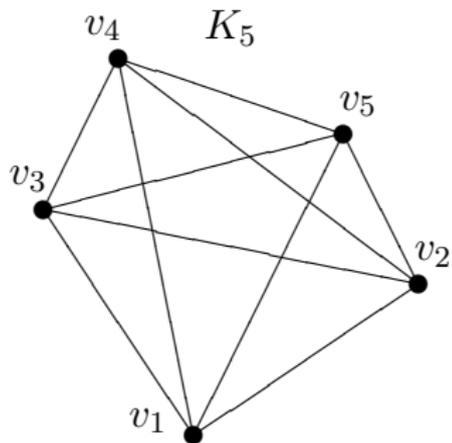
Kapitel IV Minimale Spannbäume

1. Grundlagen

Ein Graph $G = (V, E)$ besteht aus einer Menge V von Knoten und einer Menge E von Kanten. Wir werden nur endliche Knoten- (und damit auch Kanten-) Mengen betrachten. Die Anzahl der Knoten bezeichnen wir mit n ($|V| = n$), die Anzahl der Kanten mit m ($|E| = m$). Eine **gerichtete** Kante mit den Endknoten u und v wird mit (u, v) , eine **ungerichtete** mit $\{u, v\}$ notiert. Eine Kante $(v, v) \in E$ (bzw. $\{v, v\}$) heißt **Schlinge**. Falls E eine Multimenge ist, spricht man von **Mehrfachkanten**. Kanten (u, v) und (v, u) heißen **antiparallel**. Graphen ohne Schlingen und Mehrfachkanten heißen **einfache** Graphen (engl. **simple**). Für einfache ungerichtete Graphen gilt daher:

$$E \subseteq \binom{V}{2} := \{X \subseteq V, |X| = 2\}$$

Graphische Darstellung:



Ist $E \subseteq V \times V$, dann heißt G gerichteter Graph (engl. digraph).



Der zu G gehörige ungerichtete Graph ist $G' = (V, E')$. E' erhält man, indem man in E die Richtungen weglässt und Mehrfachkanten beseitigt.

Sei $v \in V$. Unter der Nachbarschaft

$N(v) := \{w; (v, w) \text{ oder } (w, v) \in E\}$ eines Knotens v versteht man die Menge der direkten Nachbarn von v . Der Grad eines Knotens ist definiert als:

$$\deg(v) = \begin{cases} |N(v)| & ; \text{ falls } G \text{ ungerichtet und} \\ \text{indeg}(v) + \text{outdeg}(v) & ; \text{ falls } G \text{ gerichtet.} \end{cases}$$

Dabei ist $\text{indeg}(v)$ die Anzahl aller Kanten, die v als Endknoten, und $\text{outdeg}(v)$ die Anzahl aller Kanten, die v als Anfangsknoten haben.

Beobachtung: Für einfache (gerichtete oder ungerichtete) Graphen gilt

$$\sum_{v \in V} \deg(v) = 2|E|.$$

Korollar 94

In jedem (einfachen) Graphen ist die Anzahl der Knoten mit ungeradem Grad eine gerade Zahl.

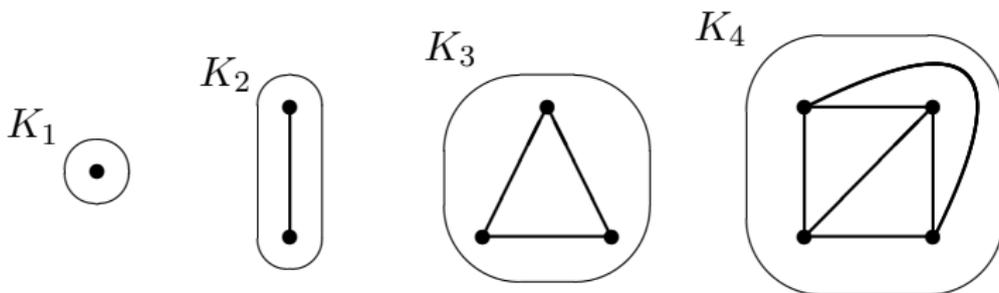
Das **Komplement** $\bar{G} = (V, \binom{V}{2} \setminus E)$ eines Graphen $G = (V, E)$ besitzt die gleiche Knotenmenge V und hat als Kantenmenge alle Kanten des vollständigen Graphen ohne die Kantenmenge E .

Ein Graph $H = (V', E')$ heißt **Teilgraph** (aka. Subgraph) von $G = (V, E)$, falls $V' \subseteq V$ und $E' \subseteq E$. H heißt **(knoten-) induzierter Teilgraph**, falls H Teilgraph von G ist und

$$E' = E \cap \binom{V'}{2}.$$

Ein **Kantenzug** (oder **Pfad**) ist eine Folge $e_1 := \{v_0, v_1\}, \dots, e_l := \{v_{l-1}, v_l\}$. v_0 und v_l sind die Endknoten, l ist die Länge des Kantenzuges. Sind bei einem Pfad alle v_i (und damit erst recht alle e_i) verschieden, so sprechen wir von einem **einfachen Pfad**. Ein Kantenzug mit $v_l = v_0$ heißt **Zykel** oder **Kreis**. Ein Kreis, in dem alle v_i verschieden sind, heißt **einfacher Kreis**.

Ein (ungerichteter) Graph G heißt **zusammenhängend**, wenn es für alle $u, v \in V$ einen Pfad gibt, der u und v verbindet. Ein (knoten-)maximaler induzierter zusammenhängender Teilgraph heißt **(Zusammenhangs-)Komponente**.



Ein Graph G heißt **azyklisch**, wenn er keinen Kreis enthält. Wir bezeichnen einen solchen ungerichteten Graphen dann als **Wald**. Ist dieser auch zusammenhängend, so sprechen wir von einem **Baum**. Ist der Teilgraph $T = (V, E') \subseteq G = (V, E)$ ein Baum, dann heißt T ein **Spannbaum** von G .

Satz 95

Ist $T = (V, E)$ ein Baum, dann ist $|E| = |V| - 1$.

Beweis:

Induktion über die Anzahl n der Knoten:

$n = 0, 1$: klar.

$n \rightarrow n + 1$: Sei $|V| = n + 1$. Da T zusammenhängend ist, ist $\deg(v) \geq 1$ für alle $v \in V$. T muss einen Knoten v mit $\deg(v) = 1$ enthalten, denn ansonsten würde, wie wiederum eine einfache Induktion zeigt, T einen Kreis enthalten. Wende nun die Induktionsannahme auf den durch $V - \{v\}$ induzierten Teilgraphen an. □

Korollar 96

Ein (ungerichteter) Graph G ist zusammenhängend, gdw G einen Spannbaum hat.

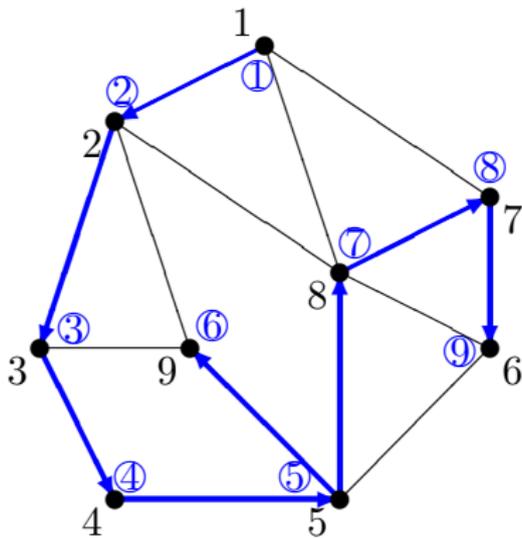
2. Traversierung von Graphen

Sei $G = (V, E)$ ein ungerichteter Graph. Anhand eines Beipfels betrachten wir die zwei Algorithmen **DFS** (Tiefensuche) und **BFS** (Breitensuche).

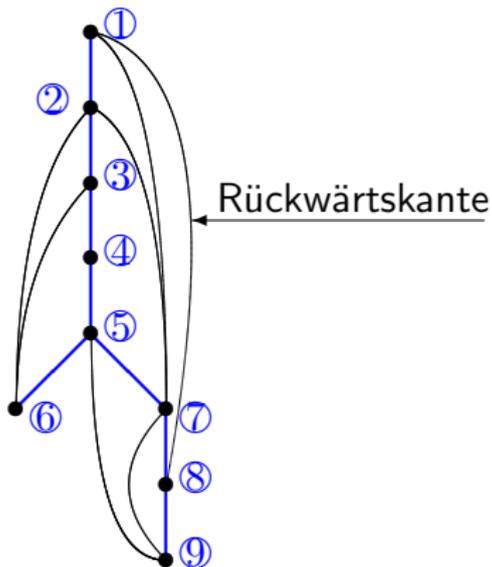
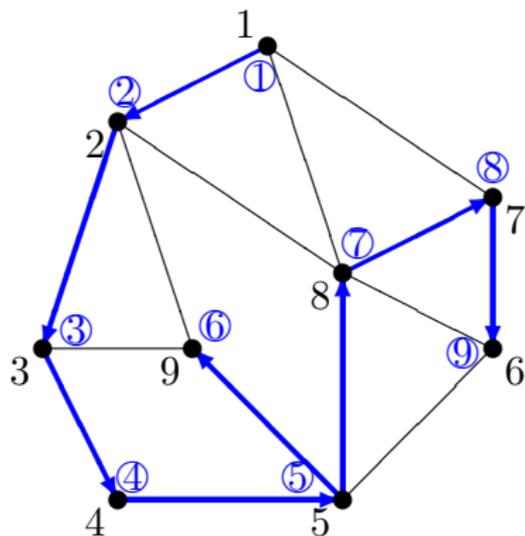
2.1 DFS-Algorithmus

```
while  $\exists$  unvisited  $v$  do  
     $r :=$  pick (random) unvisited node  
    push  $r$  onto stack  
    while stack  $\neq \emptyset$  do  
         $v :=$  pop top element  
        if  $v$  unvisited then  
            mark  $v$  visited  
            push all neighbours of  $v$  onto stack  
            perform operations DFS_Ops( $v$ )  
        fi  
    od  
od
```

Beispiel 97



Beobachtung: Die markierten Kanten bilden einen Spannbaum:



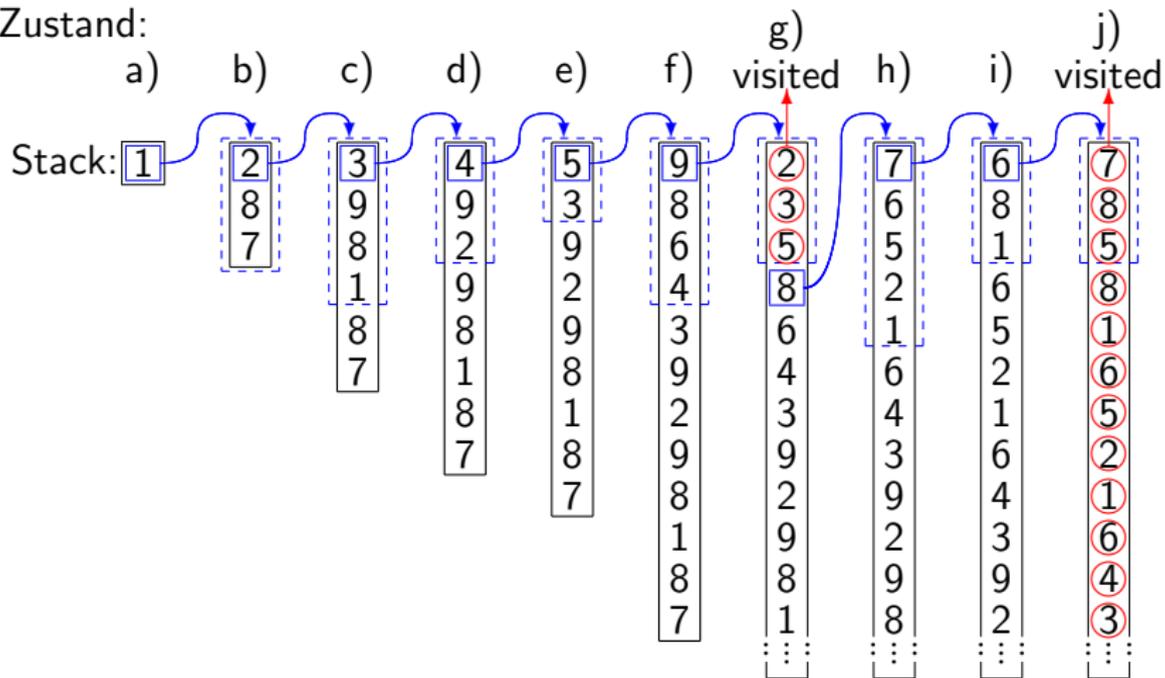
Folge der Stackzustände

□ : oberstes Stackelement

⋮ : Nachbarn

○ : schon besuchte Knoten

Zustand:



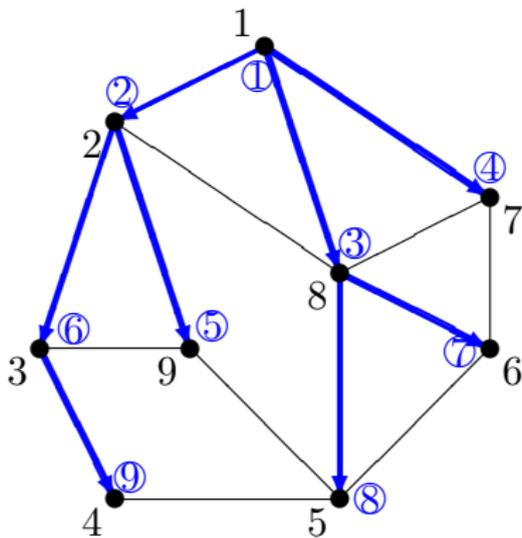
Wir betrachten den Stackzustand:

Im Zustand g) sind die Elemente 2, 3 und 5 als visited markiert (siehe Zustände b), c) und e)). Deswegen werden sie aus dem Stack entfernt, und das Element 8 wird zum obersten Stackelement. Im Zustand j) sind alle Elemente markiert, so dass eins nach dem anderen aus dem Stack entfernt wird.

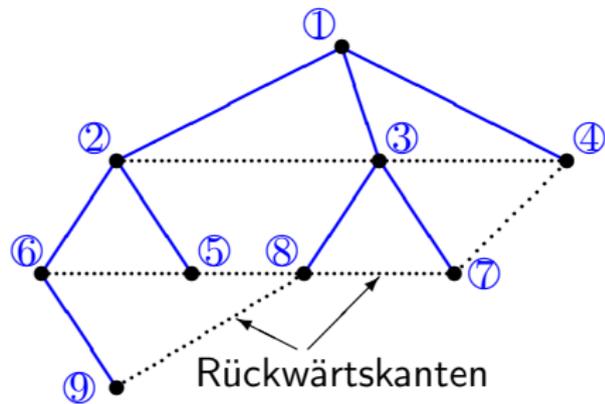
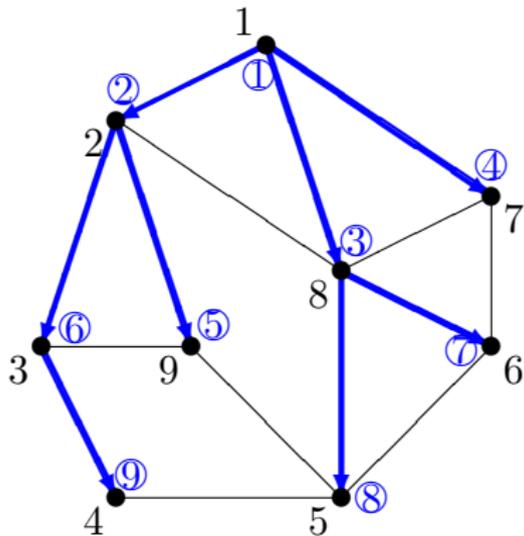
2.2 BFS-Algorithmus

```
while  $\exists$  unvisited  $v$  do  
     $r :=$  pick (random) unvisited node  
    push  $r$  into (end of) queue  
    while queue  $\neq \emptyset$  do  
         $v :=$  remove front element of queue  
        if  $v$  unvisited then  
            mark  $v$  visited  
            append all neighbours of  $v$  to end of queue  
            perform operations BFS_Ops( $v$ )  
        fi  
    od  
od
```

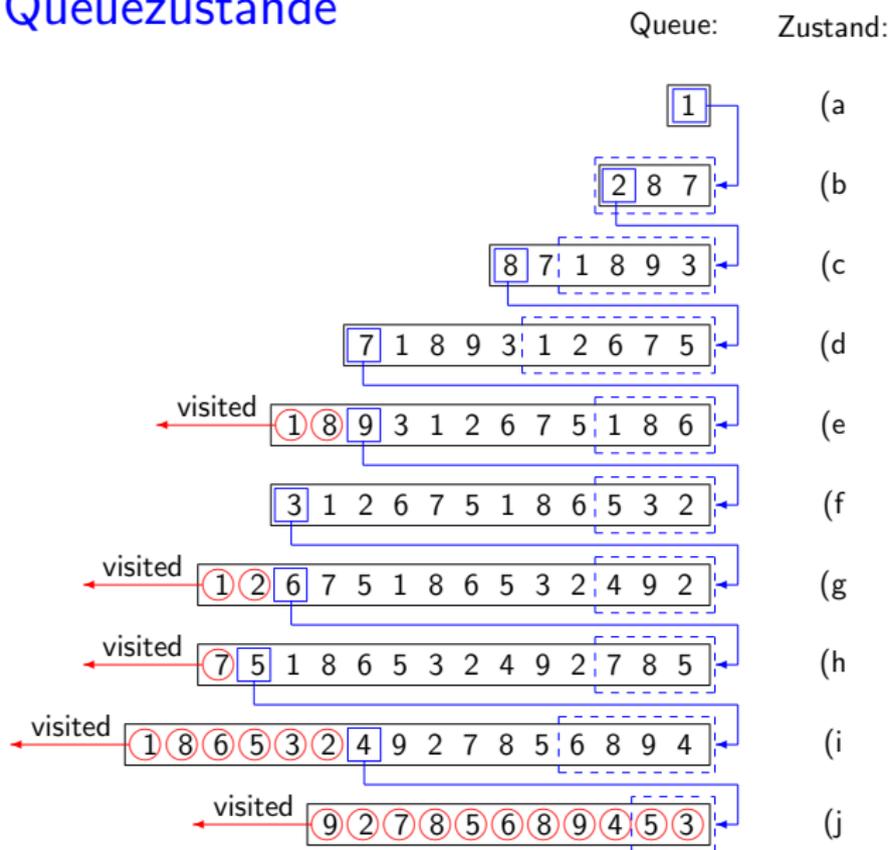
Beispiel 98



Beobachtung: Die markierten Kanten bilden einen Spannbaum:



Folge der Queuezustände



Wir betrachten die Folge der Queuezustände. Wiederum bedeutet die Notation:

-  : vorderstes Queue-Element  : Nachbarn
 : schon besuchte Knoten

Im Zustand e) sind die Elemente 1 und 8 als visited markiert (siehe Zustände a) und c)). Deswegen werden sie aus der Queue entfernt, und so wird das Element 9 das vorderste Queueelement. Das gleiche passiert in den Zuständen g), h) und i). Im Zustand j) sind alle Elemente markiert, so dass sie eins nach dem anderen aus der Queue entfernt werden.

3. Minimale Spannbäume

Sei $G = (V, E)$ ein einfacher ungerichteter Graph, der o.B.d.A. zusammenhängend ist. Sei weiter $w : E \rightarrow \mathbb{R}$ eine Gewichtsfunktion auf den Kanten von G .

Wir setzen

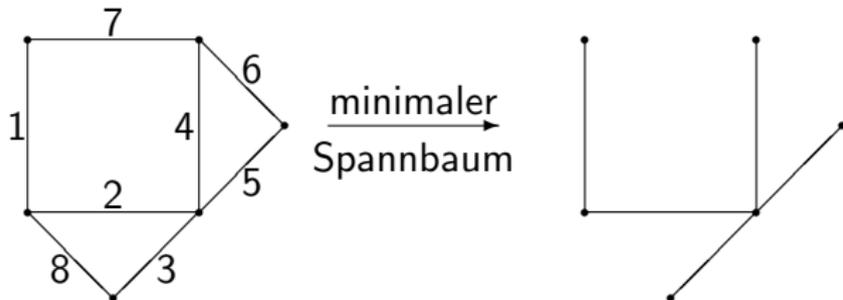
- $E' \subseteq E: w(E') = \sum_{e \in E'} w(e)$,
- $T = (V', E')$ ein Teilgraph von $G: w(T) = w(E')$.

Definition 99

T heißt **minimaler** Spannbaum (MSB, MST) von G , falls T Spannbaum von G ist und gilt:

$$w(T) \leq w(T') \text{ für alle Spannbäume } T' \text{ von } G.$$

Beispiel 100



Anwendungen:

- Telekom: Verbindungen der Telefonvermittlungen
- Leiterplatten

3.1 Konstruktion von minimalen Spann­bäumen

Es gibt zwei Prinzipien für die Konstruktion von minimalen Spann­bäumen (Tarjan):

- „blaue“ Regel
- „rote“ Regel

Satz 101

Sei $G = (V, E)$ ein zusammenhängender ungerichteter Graph, $w : E \rightarrow \mathbb{R}$ eine Gewichtsfunktion, $C = (V_1, V_2)$ ein Schnitt (d.h. $V = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, $V_1 \neq \emptyset \neq V_2$). Sei weiter $E_C = E \cap (V_1 \times V_2)$ die Menge der Kanten „über den Schnitt hinweg“. Dann gilt: („blaue“ Regel)

- 1 Ist $e \in E_C$ die **einzigste** Kante minimalen Gewichts (über alle Kanten in E_C), dann ist e in **jedem** minimalen Spannbaum für (G, w) enthalten.
- 2 Hat $e \in E_C$ minimales Gewicht (über alle Kanten in E_C), dann gibt es einen minimalen Spannbaum von (G, w) , der e enthält.

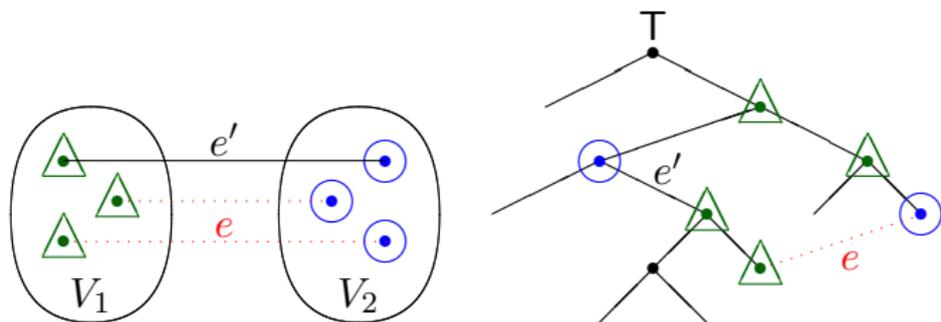
Beweis:

[durch Widerspruch]

- 1 Sei T ein minimaler Spannbaum von (G, w) , sei $e \in E_C$ die minimale Kante. Annahme: $e \notin T$. Da T Spannbaum $\Rightarrow T \cap E_C \neq \emptyset$.

Sei $T \cap E_C = \{e_1, e_2, \dots, e_k\}$, $k \geq 1$. Dann enthält $T \cup \{e\}$ einen eindeutig bestimmten Kreis (den sogenannten durch e bzgl. T bestimmten Fundamentalkreis). Dieser Kreis muss mindestens eine Kante $\in E_C \cap T$ enthalten, da die beiden Endpunkte von e auf verschiedenen Seiten des Schnitts C liegen.

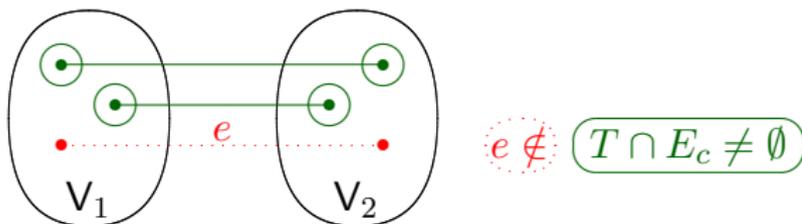
Beweis (Forts.):



Sei $e' \in E_C \cap T$. Dann gilt nach Voraussetzung $w(e') > w(e)$. Also ist $T' := T - \{e'\} \cup \{e\}$ ein Spannbaum von G , der echt kleineres Gewicht als T hat, Widerspruch zu „ T ist minimaler Spannbaum“.

Beweis (Forts.):

- ② Sei $e \in E_C$ minimal. Annahme: e kommt in **keinem** minimalen Spannbaum vor. Sei T ein beliebiger minimaler Spannbaum von (G, w) .



$e \notin T \cap E_C \neq \emptyset$. Sei $e' \in E_C \cap T$ eine Kante auf dem durch e bezüglich T erzeugten Fundamentalkreis. Dann ist $T' = T - \{e'\} \cup \{e\}$ wieder ein Spannbaum von G , und es ist $w(T') \leq w(T)$. Also ist T' minimaler Spannbaum und $e \in T'$.

□

Satz 102

Sei $G = (V, E)$ ein ungerichteter, gewichteter, zusammenhängender Graph mit Gewichtsfunktion $w : E \rightarrow \mathbb{R}$.

Dann gilt: („rote“ Regel)

- 1 Gibt es zu $e \in E$ einen Kreis C in G , der e enthält und $w(e) > w(e')$ für alle $e' \in C \setminus \{e\}$ erfüllt, dann kommt e in **keinem** minimalen Spannbaum vor.
- 2 Ist $C_1 = e_1, \dots, e_k$ ein Kreis in G und $w(e_i) = \max\{w(e_j); 1 \leq j \leq k\}$, dann gibt es einen minimalen Spannbaum, der e_i nicht enthält.

Beweis:

- 1 Nehmen wir an, dass es einen minimalen Spannbaum T gibt, der $e = \{v_1, v_2\}$ enthält. Wenn wir e aus T entfernen, so zerfällt T in zwei nicht zusammenhängende Teilbäume T_1 und T_2 mit $v_i \in T_i$, $i = 1, 2$. Da aber e auf einem Kreis in G liegt, muss es einen Weg von v_1 nach v_2 geben, der e nicht benützt. Mithin gibt es eine Kante $\hat{e} \neq e$ auf diesem Weg, die einen Knoten in T_1 mit T_2 verbindet. Verbinden wir T_1 und T_2 entlang \hat{e} , so erhalten wir einen von T verschiedenen Spannbaum \hat{T} . Wegen $w(\hat{e}) < w(e)$ folgt $w(\hat{T}) < w(T)$, im Widerspruch zur Minimalität von T .

Beweis (Forts.):

- ② Wir nehmen an, e_i liege in jedem minimalen Spannbaum (MSB) von G , und zeigen die Behauptung durch Widerspruch.

Sei T ein beliebiger MSB von G . Entfernen wir e_i aus T , so zerfällt T in zwei nicht zusammenhängende Teilbäume T_1 und T_2 . Da e_i auf einem Kreis $C_1 = e_1, \dots, e_k$ in G liegt, können wir wie zuvor e_i durch eine Kante e_j des Kreises C_1 ersetzen, die T_1 und T_2 verbindet. Dadurch erhalten wir einen von T verschiedenen Spannbaum \tilde{T} , der e_i nicht enthält. Da nach Voraussetzung $w(e_j) \leq w(e_i)$ gilt, folgt $w(\tilde{T}) \leq w(T)$ (und sogar $w(\tilde{T}) = w(T)$, da T nach Annahme ein MSB ist). Also ist \tilde{T} ein MSB von G , der e_i nicht enthält, im Widerspruch zur Annahme, e_i liege in *jedem* MSB von G .



Literatur



Robert E. Tarjan:

Data Structures and Network Algorithms

SIAM CBMS-NSF Regional Conference Series in Applied
Mathematics Bd. 44 (1983)