

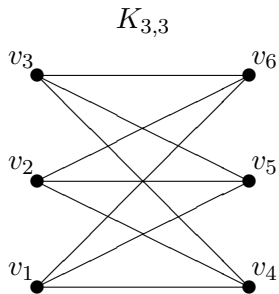
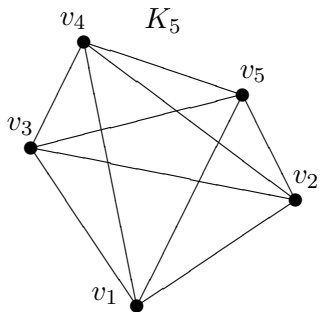
Kapitel IV Minimale Spannbäume

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 93

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

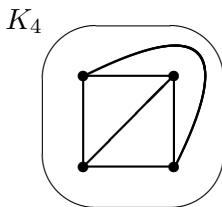
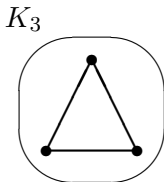
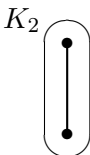
Das **Komplement** $\bar{G} = (V, \binom{V}{2} \setminus E)$ eines Graphen $G = (V, E)$ besitzt die gleiche Knotenmenge V und hat als Kantenmenge alle Kanten des vollständigen Graphen ohne die Kantenmenge E .

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 94

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 95

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

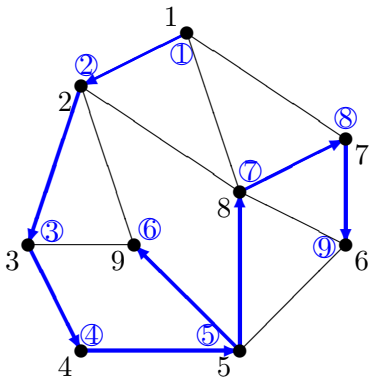
2. Traversierung von Graphen

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

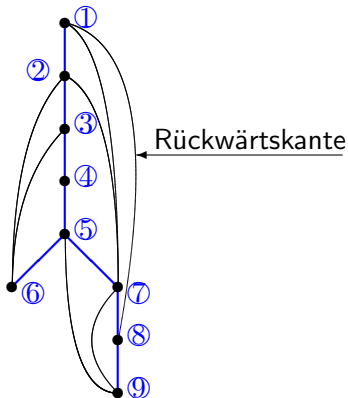
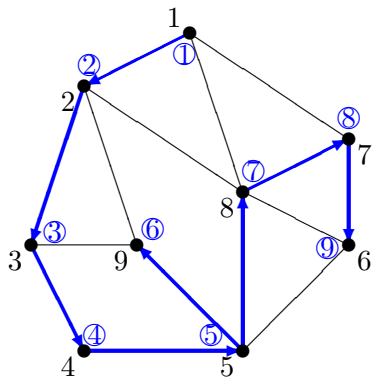
2.1 DFS-Algorithmus

```
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 96



Beobachtung: Die markierten Kanten bilden einen Spannbaum:



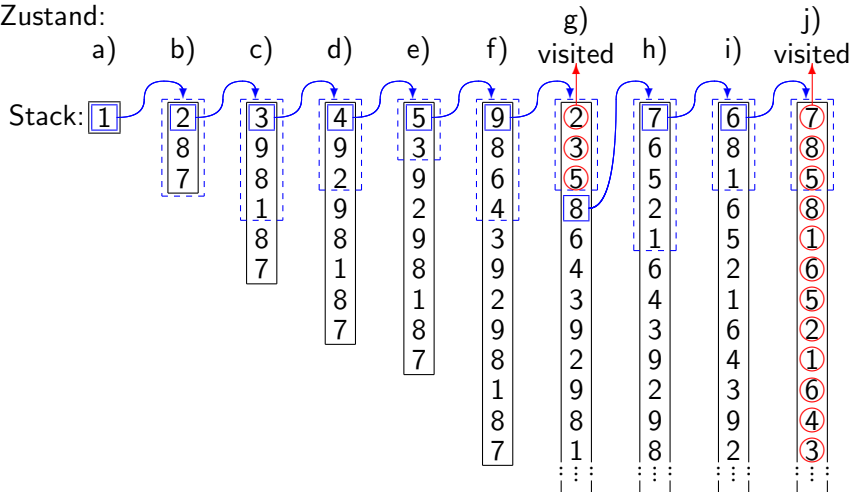
Folge der Stackzustände

□ : oberstes Stackelement

□ : Nachbarn

○ : schon besuchte Knoten

Zustand:



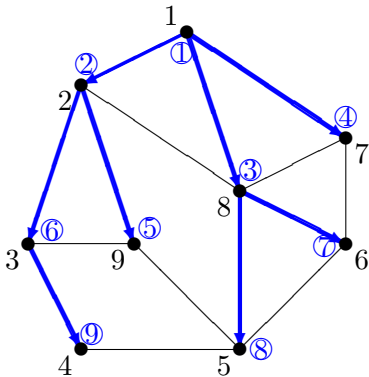
Wir betrachten den Stackzustand:

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.

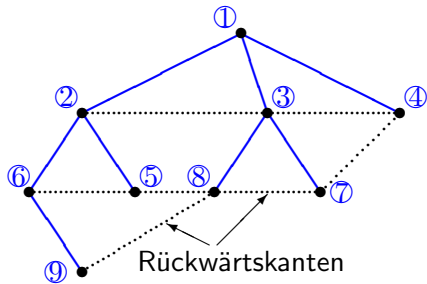
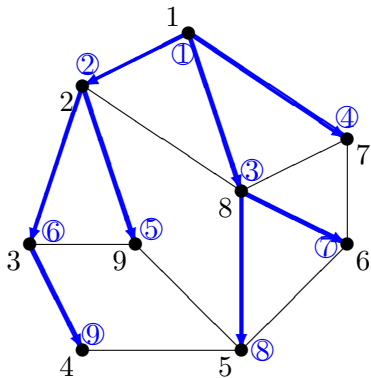
2.2 BFS-Algorithmus

```
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
```

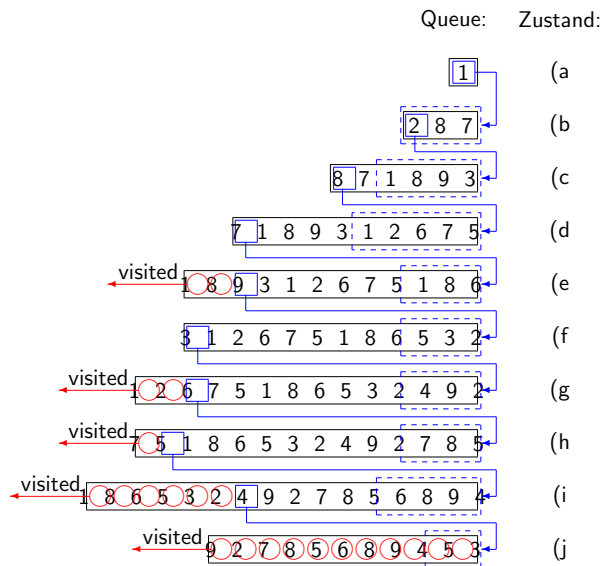
Beispiel 97






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

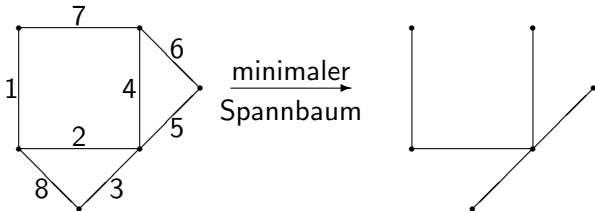
- $E' \subseteq E: w(E') = \sum_{e \in E'} w(e)$,
- $T = (V', E')$ ein Teilgraph von $G: w(T) = w(E')$.

Definition 98

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 99



Anwendungen:

- Telekom: Verbindungen der Telefonvermittlungen
- Leiterplatten