operations research
QUANTITATIVE INFRASTRUKTURSYSTEMMODELLIERUNG

Tutorium

Operations Research – Grundlagen

Technische Universität Berlin
Fachgebiet **W**irtschafts- und **I**nfrastruktur **P**olitik
WIP

Tutorium 5

Einführung Graphentheorie, Spannbäume und Traveling Salesman Problem

Tutoriumsaufgaben:

3.2	Kruskal Algorithmus: Der befreundete Busunternehmer
3.1	Minimale Spannbäume
2.33	Minimaler Spannbaum
2.34	Travelling Salesman Problem

Freiwillige Hausaufgabe:

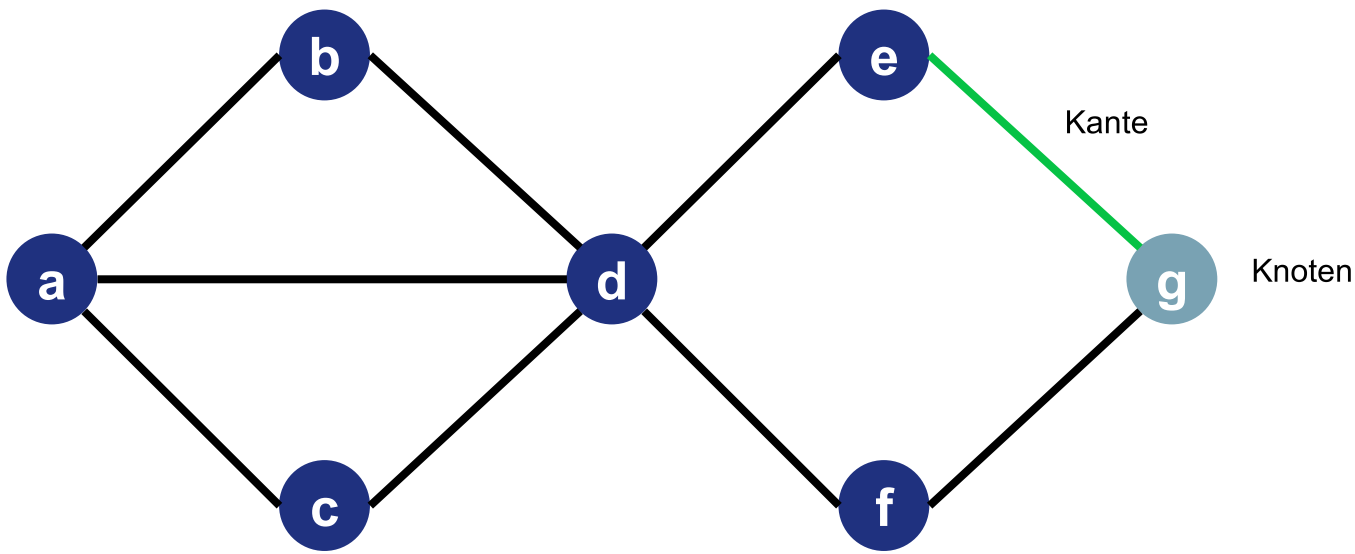
3.3	Lok um Lok
3.4	Kruskal Algorithmus: Internet in der Lausitz

Graph

Die Graphentheorie ist ein Teilgebiet der Mathematik, das die Eigenschaften von Graphen und ihre Beziehungen zueinander untersucht

Definition

Ein **Graph** *g(V,E)* ist eine Menge von Punkten *V*, die eventuell durch Kanten *E* miteinander verbunden sind, wobei es nicht auf die Form ankommt. Punkte nennt man auch Ecken oder Knoten.



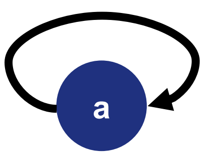
Arten von Graphen

Punkte nennt man auch Ecken oder Knoten (engl.: vertex, node)

Ungerichtete Kante – undirected edge Gerichtete Kante – directed edge

parallele Kanten (wenn gerichtet: in die gleiche Richtung)

Schlinge



Definition

Ein Graph dessen Mengen der Knoten und Kanten endlich ist, heißt **endlicher Graph**.

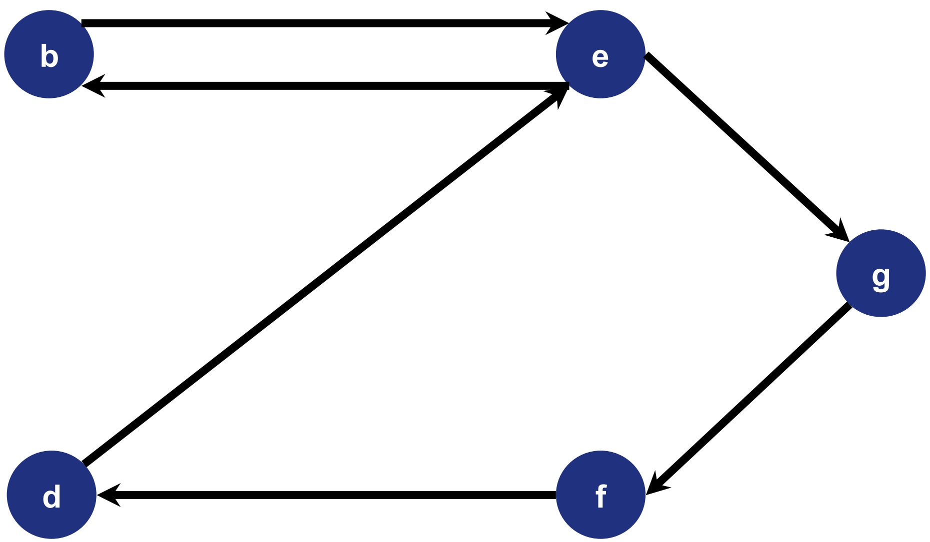
Schlichte Graphen und Digraph

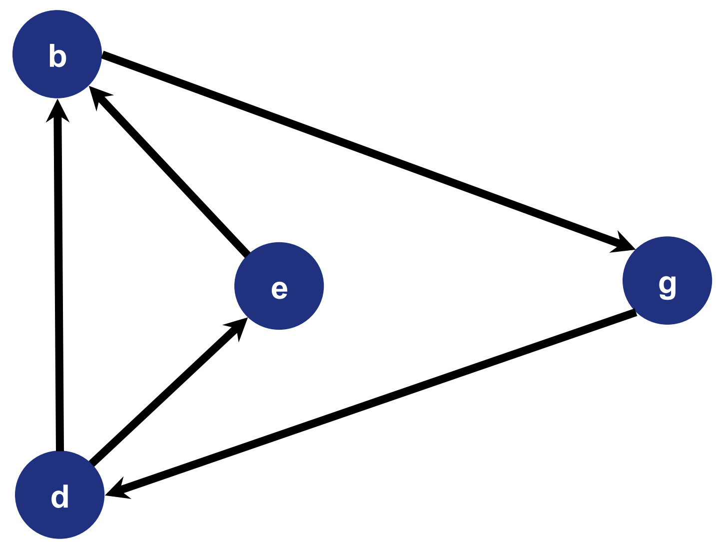
Definition

Ein Graph ohne parallele Kanten und ohne Schlinge wird als **schlichter Graph** bezeichnet.

Definition

Ein schlichter, gerichteter Graph mit endlicher Knotenmenge heißt **Digraph**.





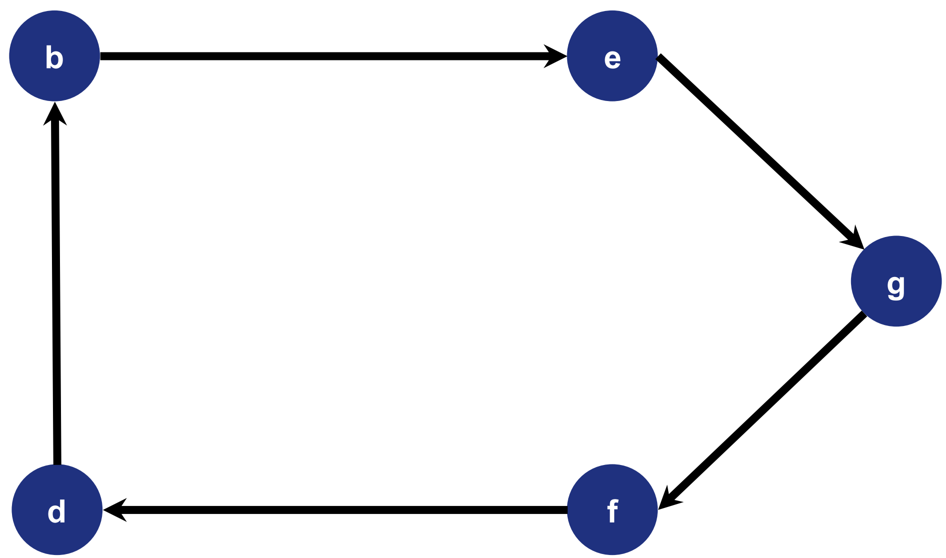
Multigraph und Zyklus

Definition

Ein Graph mit parallelen Kanten wird als **Multigraph** bezeichnet.

Definition

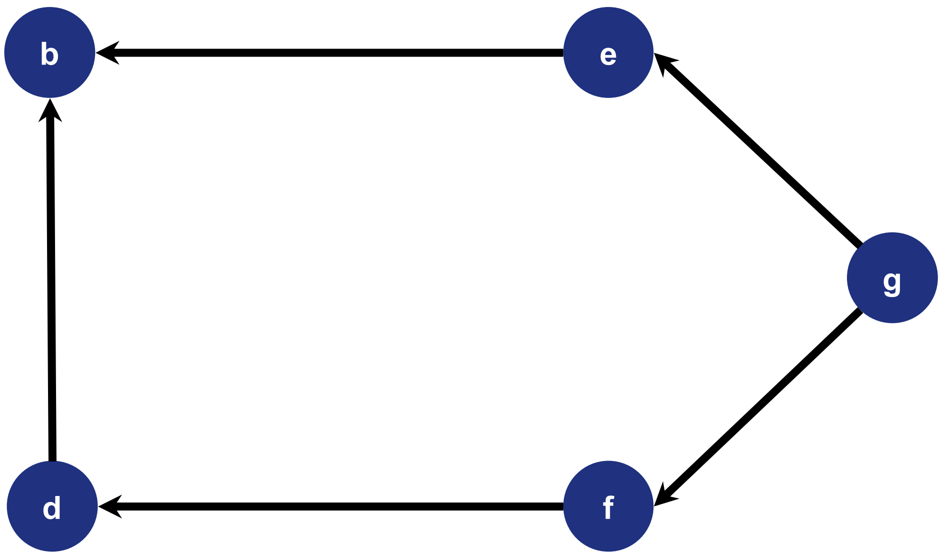
Ein **Weg** ist die Folge von gerichteten Kanten, jeweils vom Pfeilende zur Pfeilspitze. Ein geschlossener Weg heißt **Zyklus**.

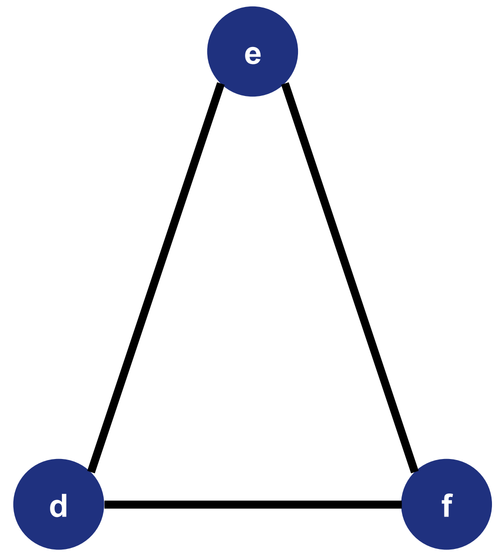


Kreis

Definition

Eine **Kette** ist die Folge von ungerichteten Kanten oder gerichteten, wobei der Richtungssinn keine Rolle spielt. Eine geschlossene Kette heißt **Kreis**.

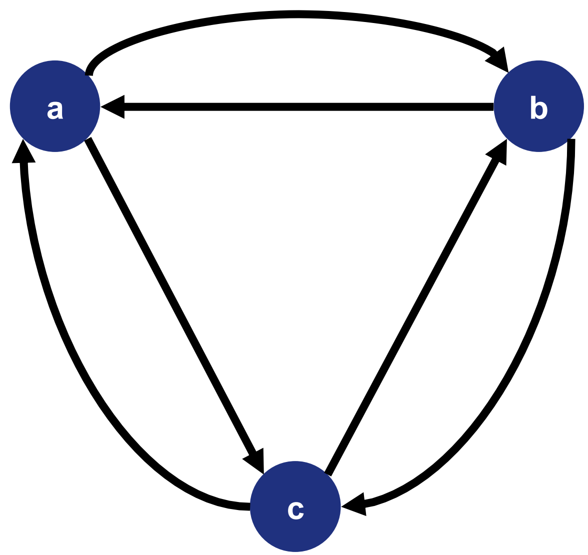




Vollständiger Graph

Definition

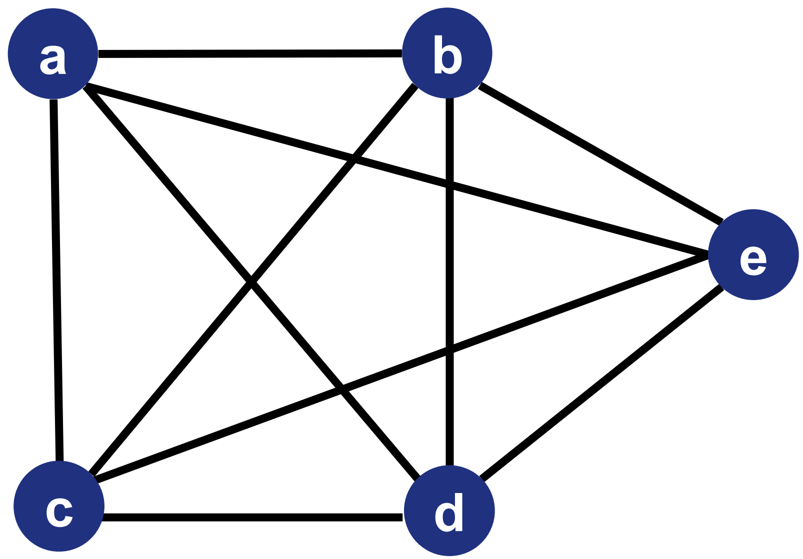
Ein Digraph heißt **vollständiger Digraph**, wenn für jedes Knotenpaar *i, j* ein Pfeil von *i* nach *j* und ein Pfeil von *j* nach *i* vorhanden ist.



Vollständiger schlichter Graph

Definition

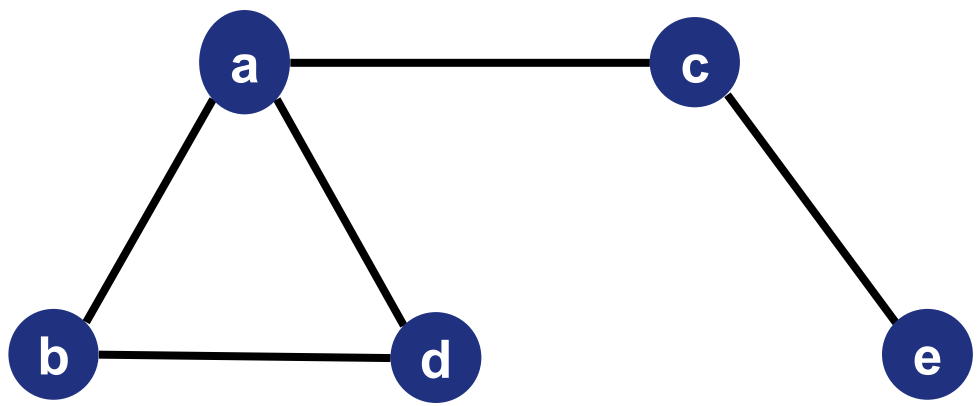
Ein ungerichteter schlichter Graph heißt **vollständiger schlichter Graph**, wenn für jedes Knotenpaar *i, j* eine Kante von *i* nach *j* existiert



Zusammenhängender Graph

Definition

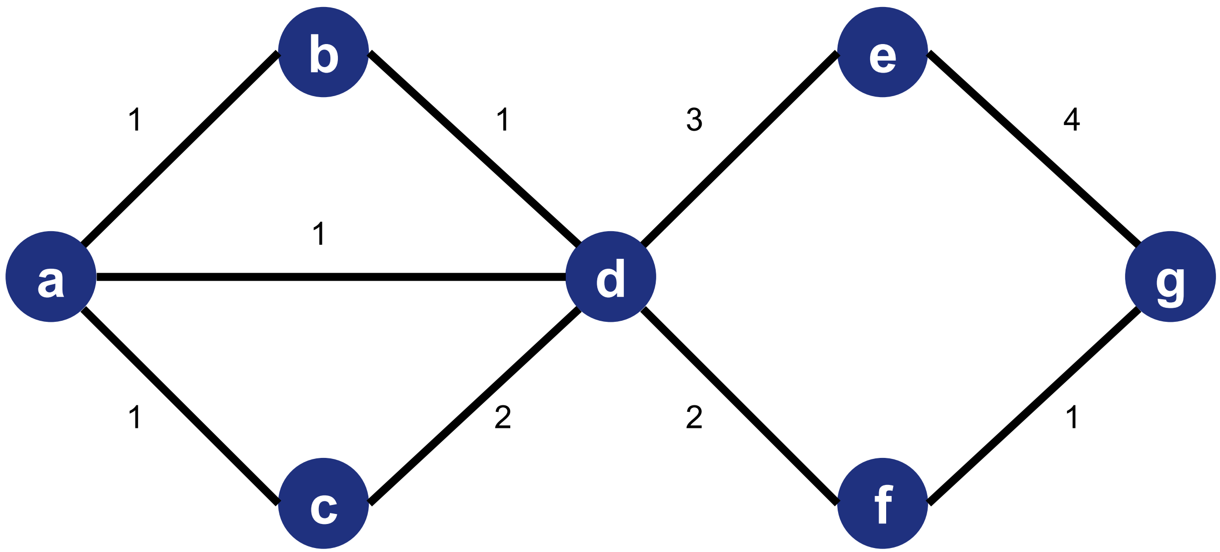
Ein Graph heißt **zusammenhängender Graph**, wenn jedes Knotenpaar durch mindestens eine Kette verbunden ist.



Gewichteter Graph

Definition

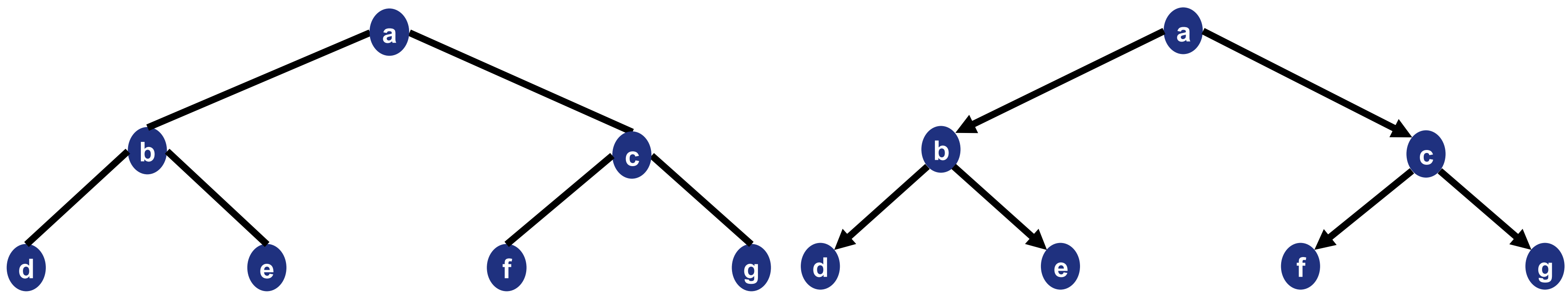
Ein Graph *g(V,E,w)* wird als **gewichteter Graph** bezeichnet, wenn alle seine Kanten *E* mit Werten *w(e)* versehen sind. Die Summe aller Pfeil-bewertungen eines Weges wird auch als Länge des Weges bezeichnet.



Baum

Definition

Ein **ungerichteter Baum** ist ein zusammenhängender, kreisfreier, ungerichteter Graph. Ein **gerichteter Baum** ist ein gerichteter, kreisfreier Graph mit genau einem Ausgangsknoten.



ungerichteter Baum

gerichteter Baum

Spannbäume

Definition

Gegeben sei ein ungerichteter Graph g mit Kantengewichten, *g(V,E,w)*. Ein **Spannbaum** von *(V,E,w)* ist ein Subgraph *(V',E')* von *(V,E)* für den gilt:

- *V'=V*, d.h. der Subgraph umfasst alle Knoten von g,
- *(V',E')* ist ein Baum (zusammenhängend und kreisfrei)

Definition

Das **Gewicht** *l(V',E')* eines Spannbaumes ist die Summe aller seiner Kantengewichte:

$l(V',E') = \sum_{e \in E'} w(e)$

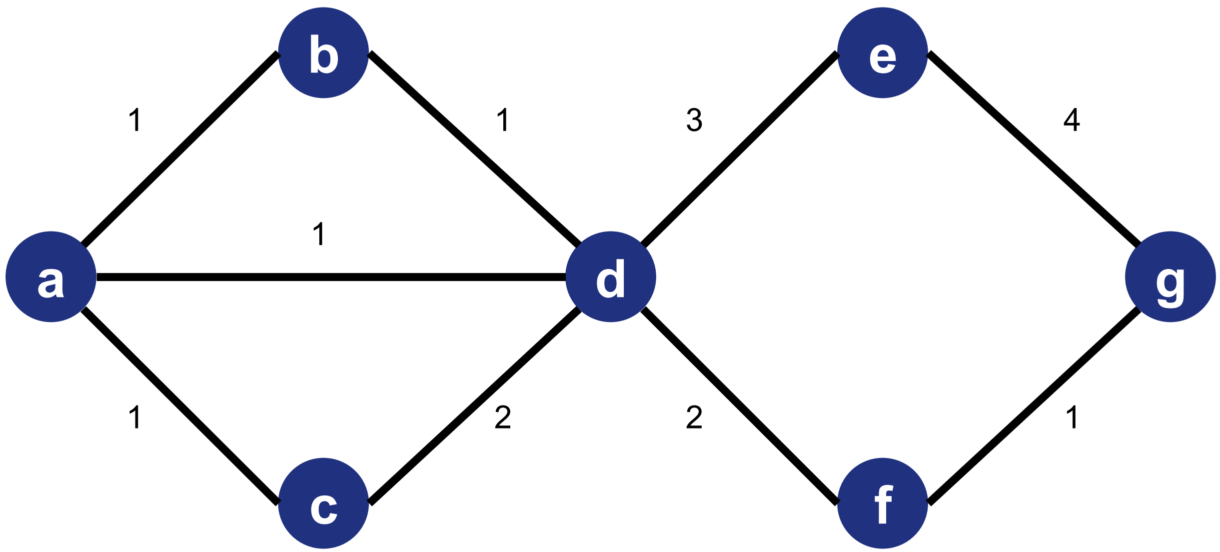
Definition

Ein **minimaler Spannbaum** hat minimales Gewicht unter allen möglichen Spannbäumen.

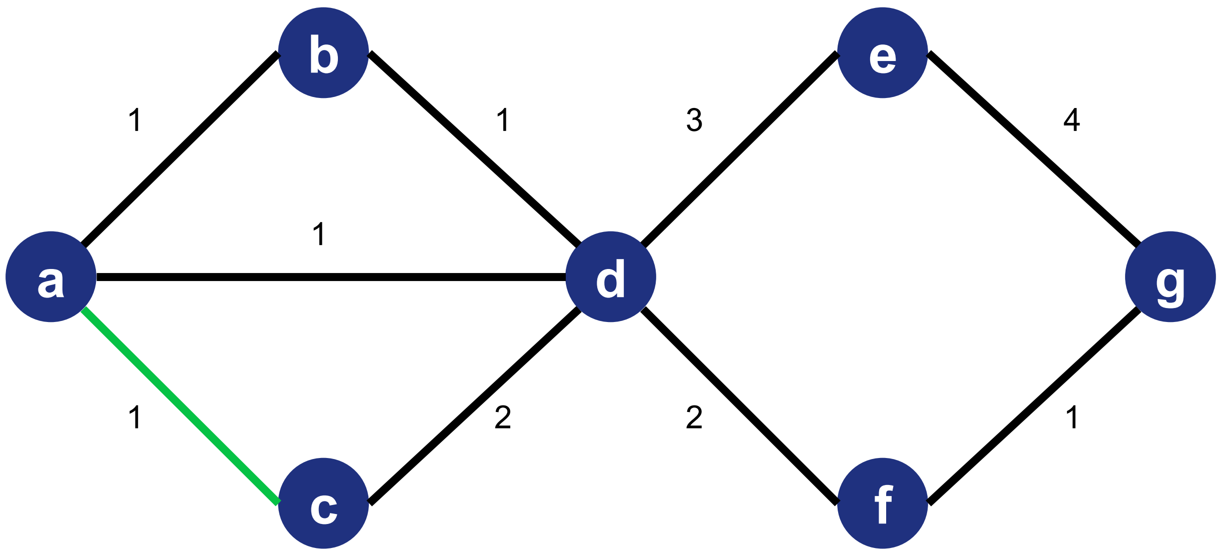
Kruskal-Algorithmus

- Joseph Kruskal 1956
- Voraussetzungen
  - Graph muss
    - endlich,
    - zusammenhängend,
    - schlingenfrei,
    - ungerichtet,
    - und gewichtet sein
- Funktionsweise
  - Wähle unter allen noch nicht ausgewählten Kanten, die Kante mit dem geringsten Gewicht, die mit den bisher markierten Kanten keinen Kreis bildet.
    - Fertig, wenn alle Knoten mit einander verbunden sind (≙ Anzahl markierter Kanten = Anzahl Knoten – 1)
    - Gewicht des minimalen Spannbaums angeben

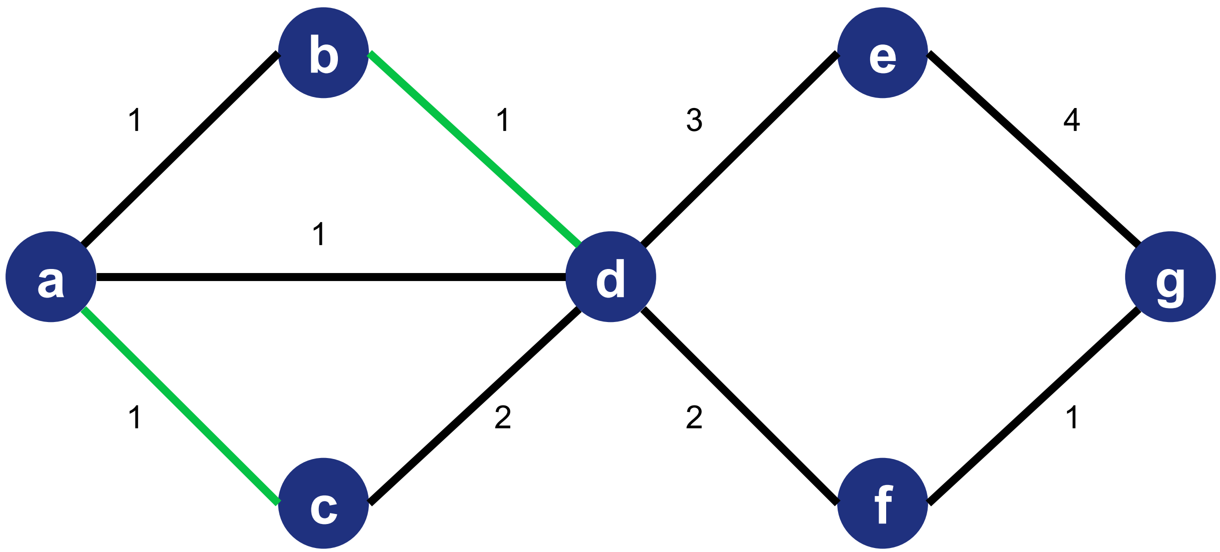
Kruskal-Algorithmus



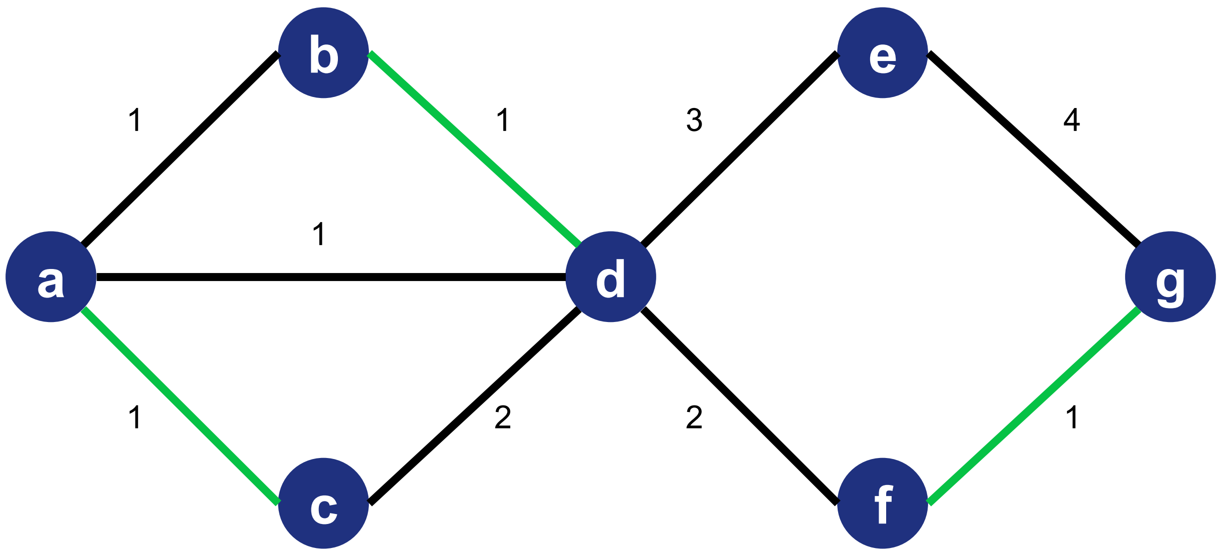
Kruskal-Algorithmus



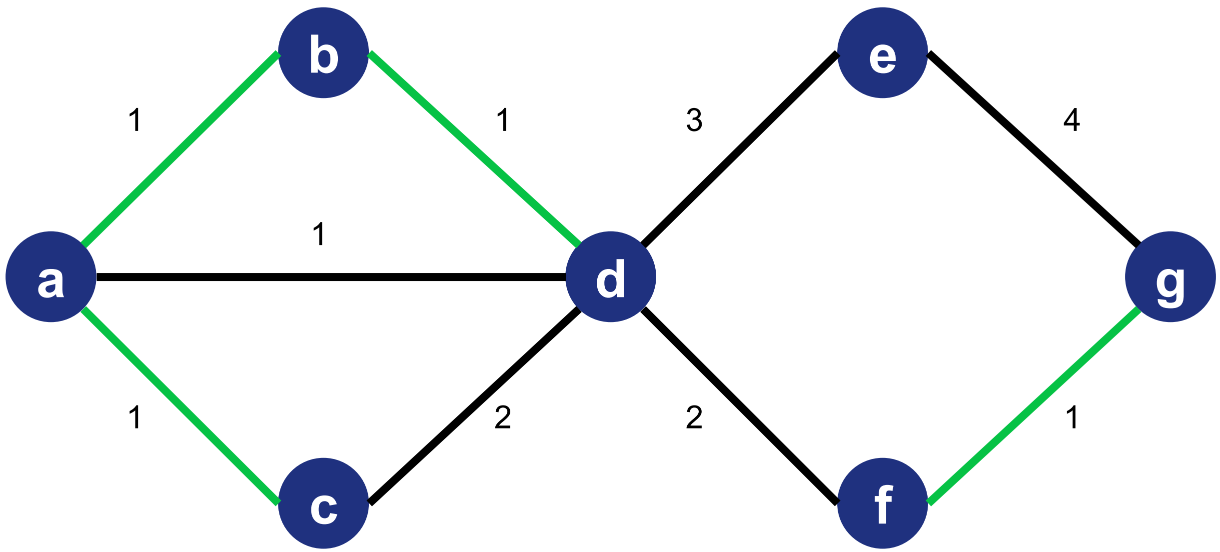
Kruskal-Algorithmus



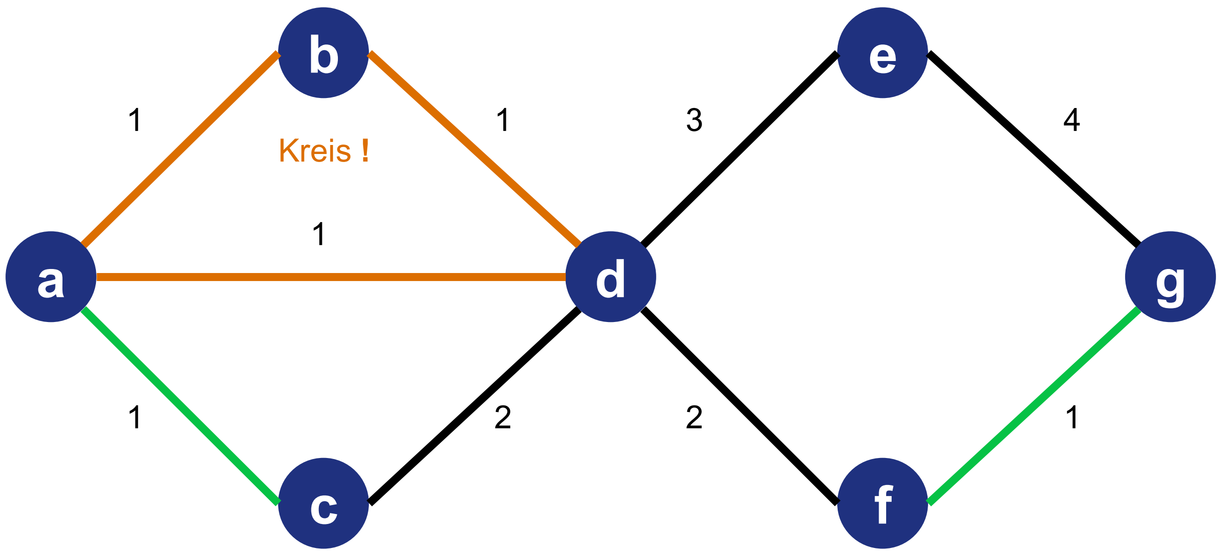
Kruskal-Algorithmus



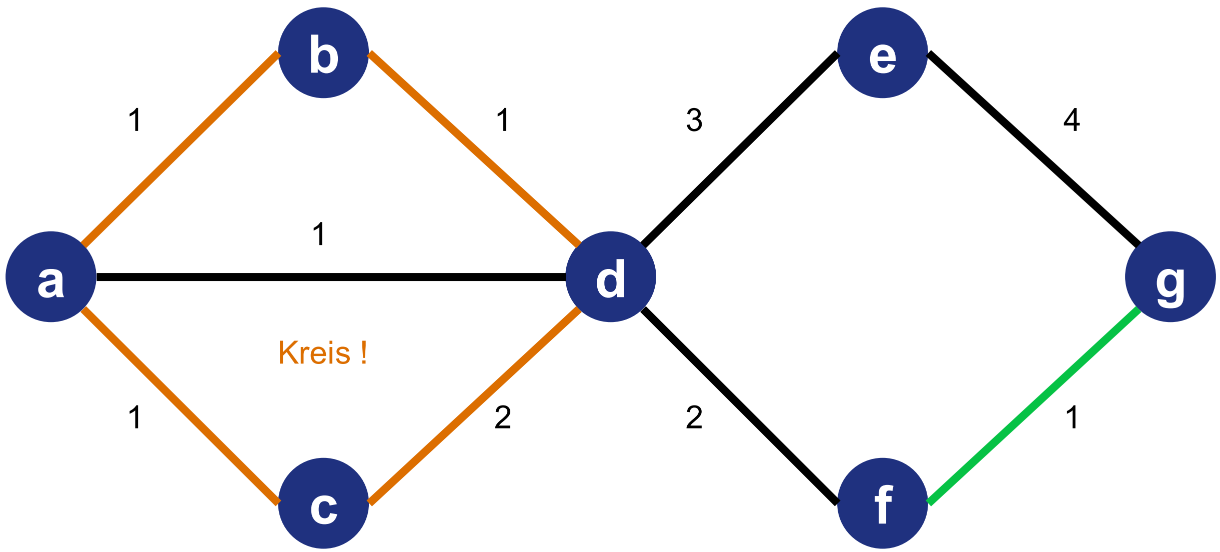
Kruskal-Algorithmus



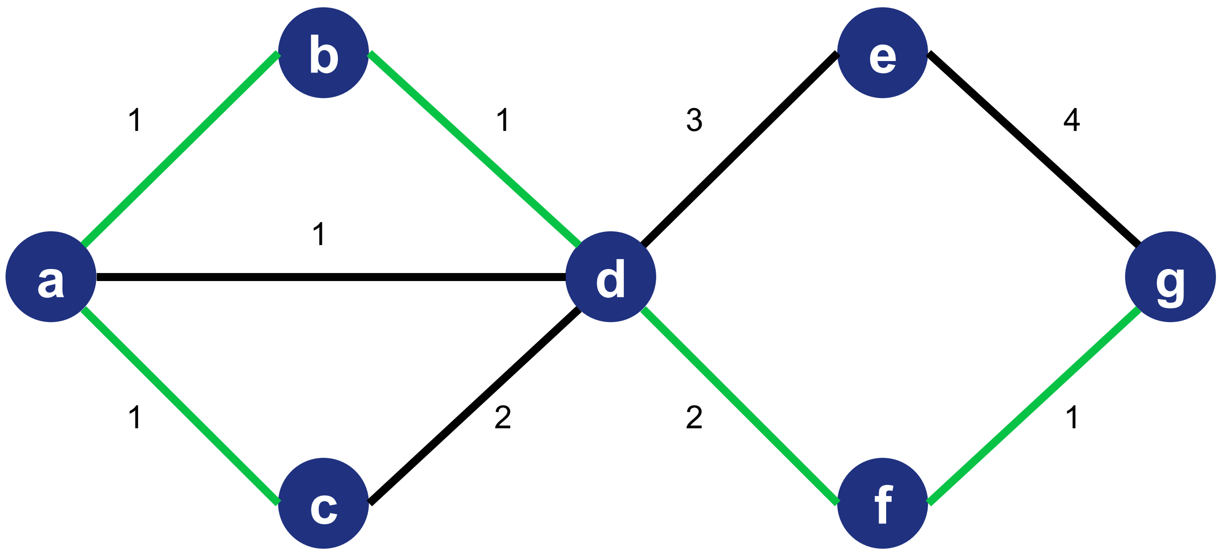
Kruskal-Algorithmus



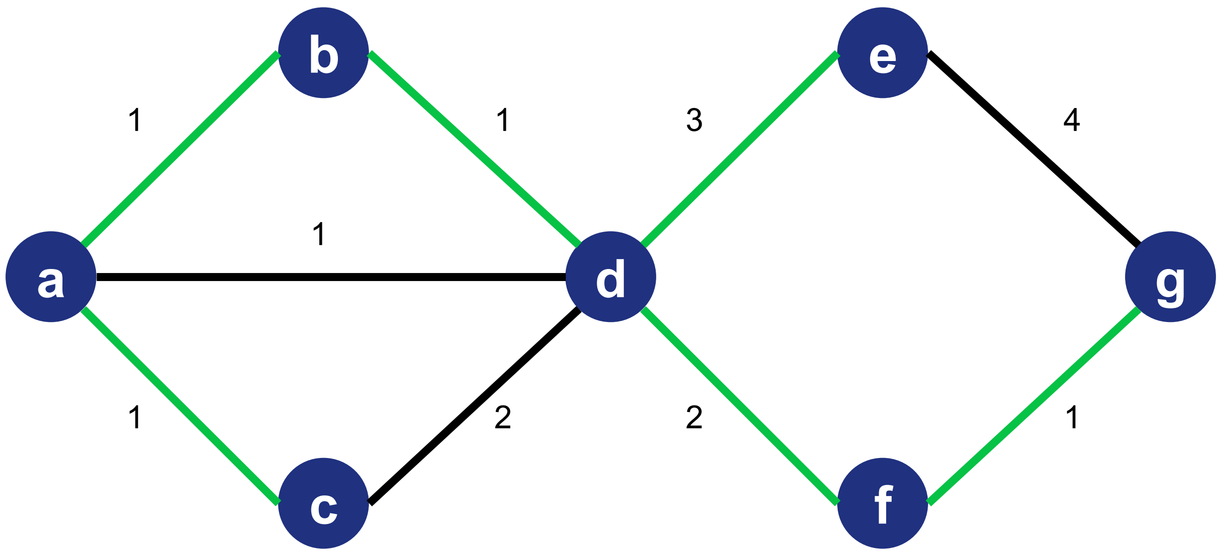
Kruskal-Algorithmus



Kruskal-Algorithmus



Kruskal-Algorithmus



Minimaler Spannbaum,
Gewicht: 9

3.2 Kruskal-Algorithmus: Der befreundete Busunternehmer

Ein Freund von Ihnen besitzt ein Busunternehmen in Berlin. Ihr Freund hat das Gefühl, dass sein kleines Busnetz nicht optimal ausgelegt ist und bittet Sie um Hilfe. Sie können davon ausgehen, dass sich die Kostenfunktion proportional zur Fahrzeit verhält. Bei einem Bierchen zeichnet Ihr Freund Ihnen den folgenden Graphen auf einen Bierdeckel, um sein Problem zu illustrieren. Die Entfernungen sind in Kilometern angegeben. Helfen Sie Ihrem Freund und finden Sie einen minimalen Spannbaum

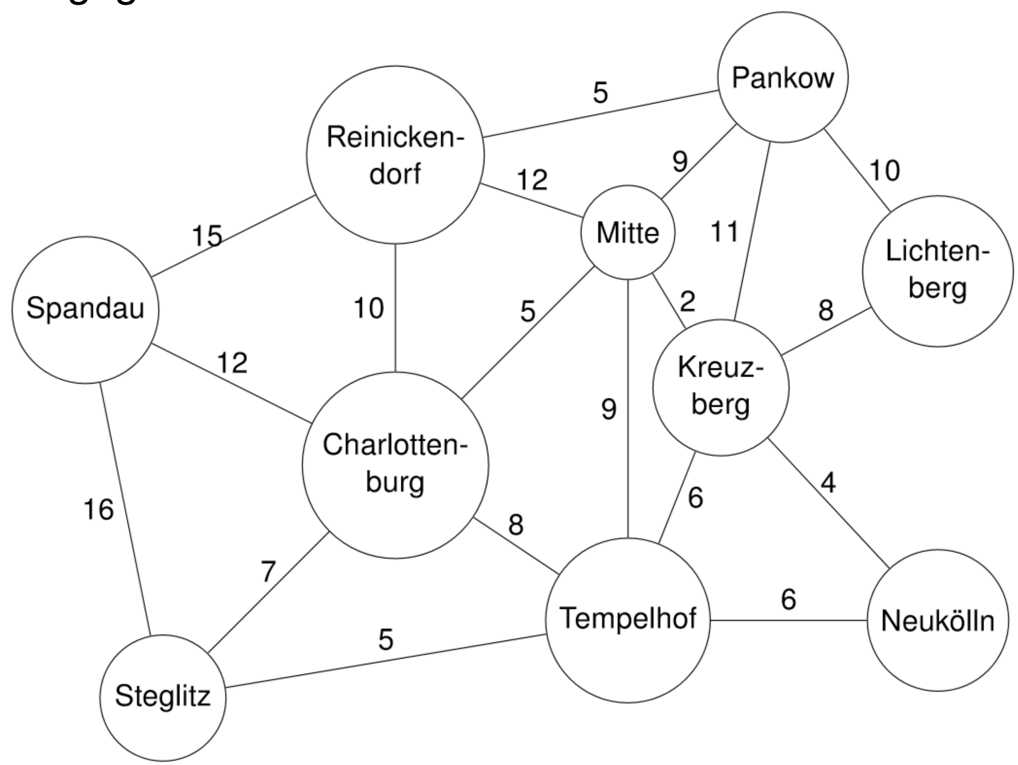


Abbildung 5: Bierdeckelzeichnung in km

3.2 Kruskal-Algorithmus: Der befreundete Busunternehmer

… finden Sie einen minimalen Spannbaum.

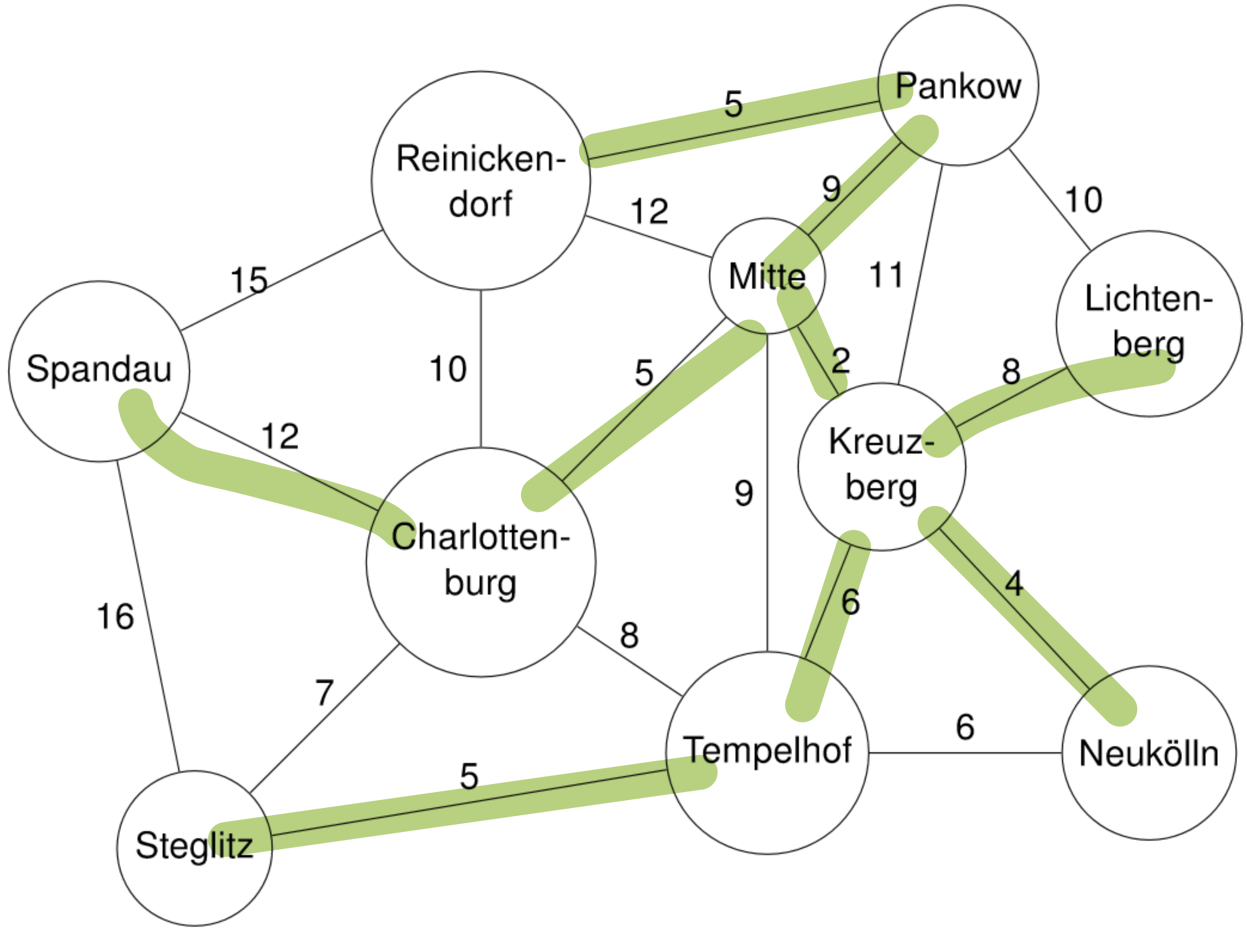


Abbildung 5: Bierdeckelzeichnung in km

3.2 Kruskal-Algorithmus: Der befreundete Busunternehmer

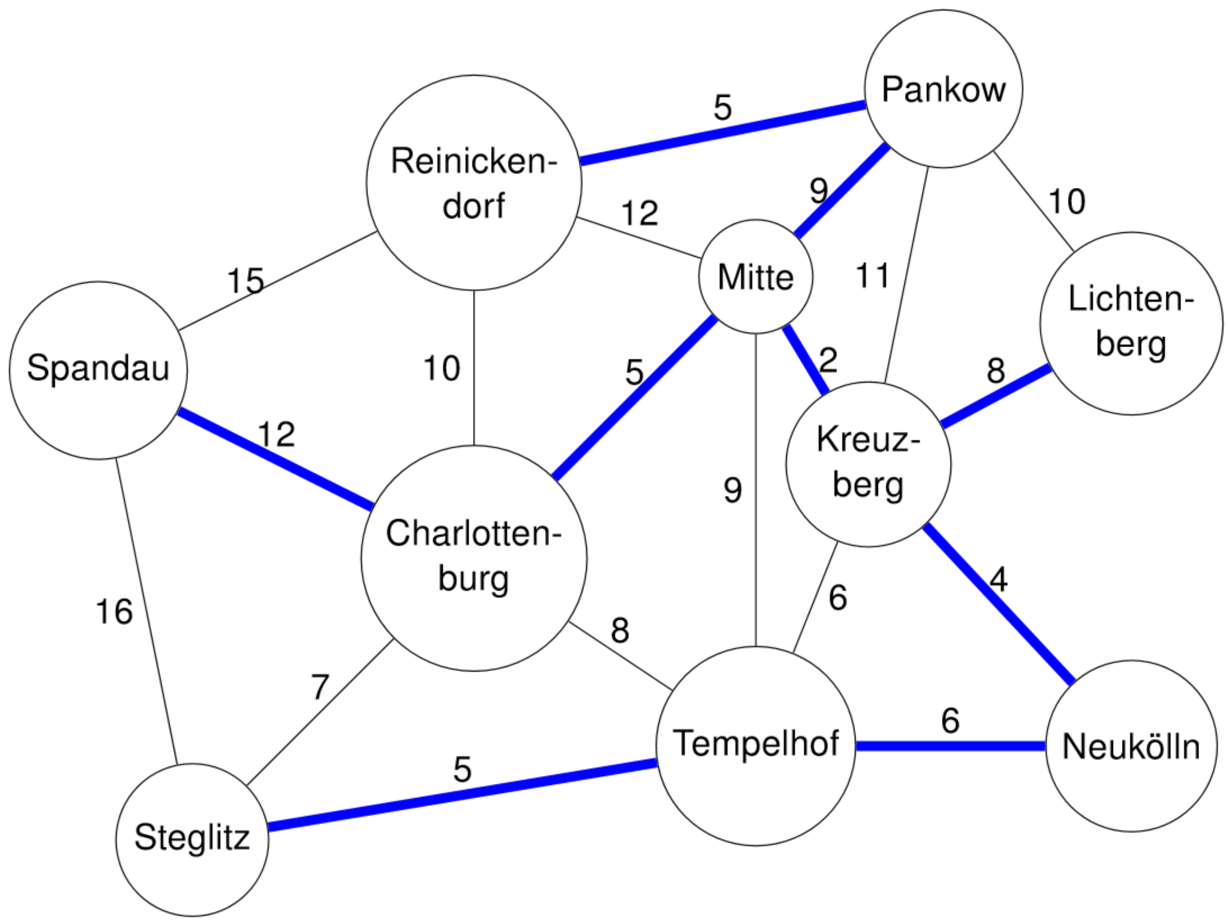
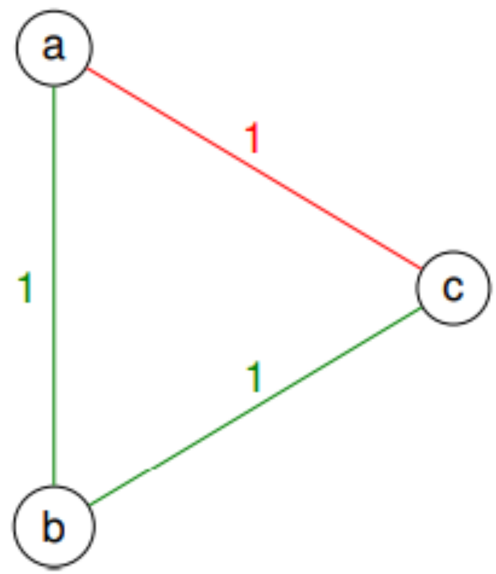


Abbildung 5: Ungerichteter Graph

3.1 Minimale Spannbäume

a) Sei g(V, E,w) ein Graph und sei e eine Kante mit minimalem Gewicht. Ist die Kante e dann immer im zugehörigen minimalen Spannbaum enthalten?

- Nein!
- Gegenbeispiel →



3.1 Minimale Spannbäume

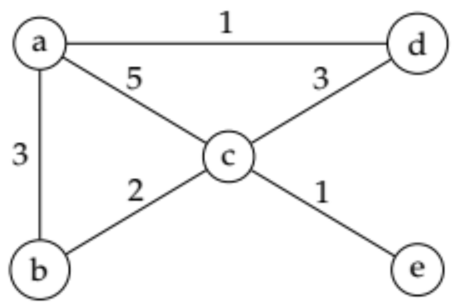
b) Kann man Kruskals Algorithmus auch verwenden, um einen maximalen Spannbaum zu berechnen?

- Ja!
- Um einen maximalen anstelle eines minimalen Spannbaums zu finden, alle Kantengewichte mit -1 multiplizieren
- Kanten mit dem größten Gewicht verhalten sich nun wie Kanten mit dem kleinsten Gewicht

2.33 Minimale Spannbäume

a) Stellen Sie das Problem des minimalen Spannbaums allgemein als Optimierungsproblem dar.

b) Nun stellen sie das spezielle Problem für den gegebenen Graphen auf.



2.33 Minimale Spannbäume

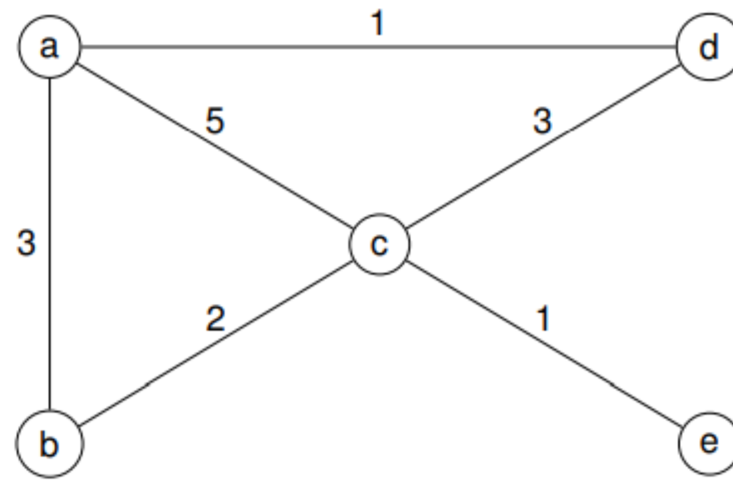
a) Stellen Sie das Problem des minimalen Spannbaums allgemein als Optimierungsproblem dar.

- Sei $g(V, E, w)$ ein Graph mit n Knoten. Für Teilmengen $S \subseteq V$ von Knoten bezeichne E(S) die Menge der Kanten, deren (beide) Endpunkte in S liegen.

$$
\begin{array}{rrccl}
\min z = & \sum_{(i,j)\in E} w(i,j)\cdot x(i,j) & & & \\
& \sum_{(i,j)\in E} x(i,j) & = & n-1 & \\
& \sum_{(i,j)\in E(S)} x(i,j) & \leq & |S|-1 & \text{fuer jede Teilmenge } S \subseteq V \\
& w(i,j) & \in & \mathbb{R} & \\
& x(i,j) & \in & \{0,1\} &
\end{array}
$$

2.33 Minimale Spannbäume

b. Nun stellen sie das spezielle Problem für den gegebenen Graphen auf.

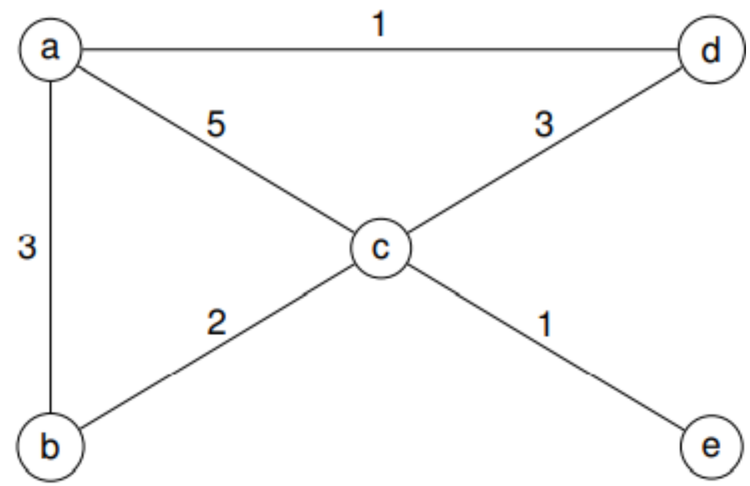


$$\min z = 3x(a,b) + 5x(a,c) + 1x(a,d) + 2x(b,c) + 3x(c,d) + x(c,e)$$
$$s.t. \quad x(i,j) \in \{0,1\}$$

2.33 Minimale Spannbäume

b. Nun stellen sie das spezielle Problem für den gegebenen Graphen auf.

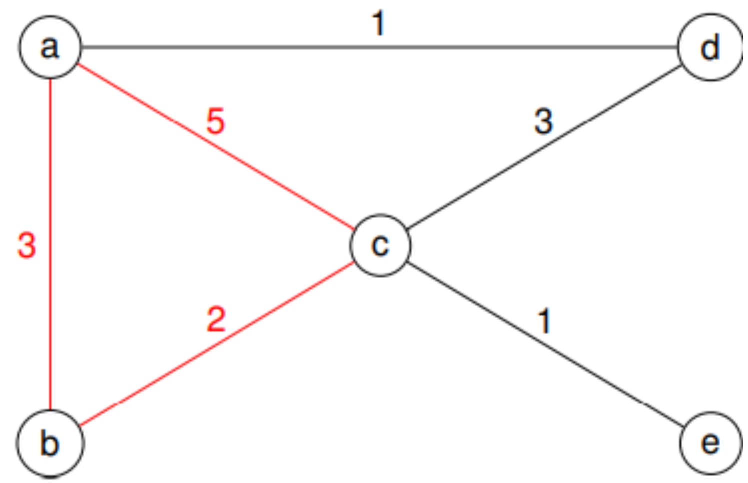
$$\min z = 3x(a,b) + 5x(a,c) + 1x(a,d) + 2x(b,c) + 3x(c,d) + x(c,e)$$
$$s.t. \quad x(a,b) + x(a,c) + x(a,d) + x(b,c) + x(c,d) + x(c,e) = 4$$
$$x(i,j) \in \{0,1\}$$



2.33 Minimale Spannbäume

b. Nun stellen sie das spezielle Problem für den gegebenen Graphen auf.

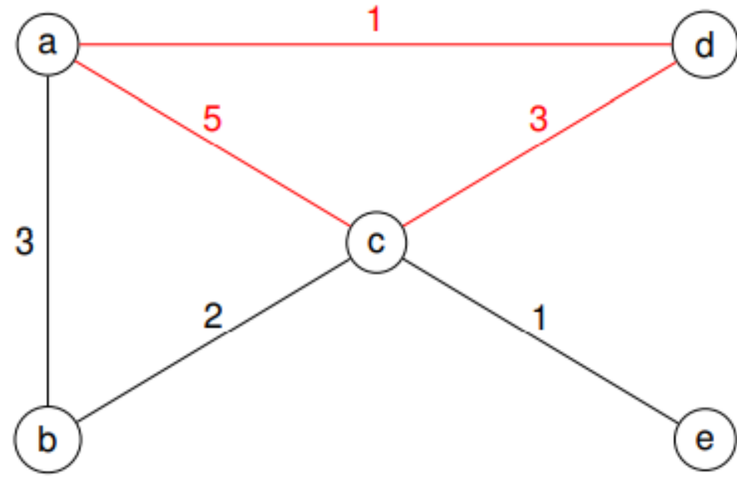
$$\begin{aligned} \min z = 3x(a,b) + 5x(a,c) + 1x(a,d) + 2x(b,c) + 3x(c,d) + x(c,e) \\ s.t. \quad x(a,b) + x(a,c) + x(a,d) + x(b,c) + x(c,d) + x(c,e) = 4 \\ x(a,b) + x(a,c) + x(b,c) \leq 2 \\ x(i,j) \in \{0,1\} \end{aligned}$$



2.33 Minimale Spannbäume

b. Nun stellen sie das spezielle Problem für den gegebenen Graphen auf.

$$\begin{aligned}
\min z = 3x(a,b) + 5x(a,c) + 1x(a,d) + 2x(b,c) + 3x(c,d) + x(c,e) \\
s.t. \quad x(a,b) + x(a,c) + x(a,d) + x(b,c) + x(c,d) + x(c,e) = 4 \\
x(a,b) + x(a,c) + x(b,c) \leq 2 \\
x(a,c) + x(a,d) + x(c,d) \leq 2 \\
x(i,j) \in \{0,1\}
\end{aligned}$$



2.33 Minimale Spannbäume

b. Nun stellen sie das spezielle Problem für den gegebenen Graphen auf.

$$\min z = 3x(a,b) + 5x(a,c) + 1x(a,d) + 2x(b,c) + 3x(c,d) + x(c,e)$$

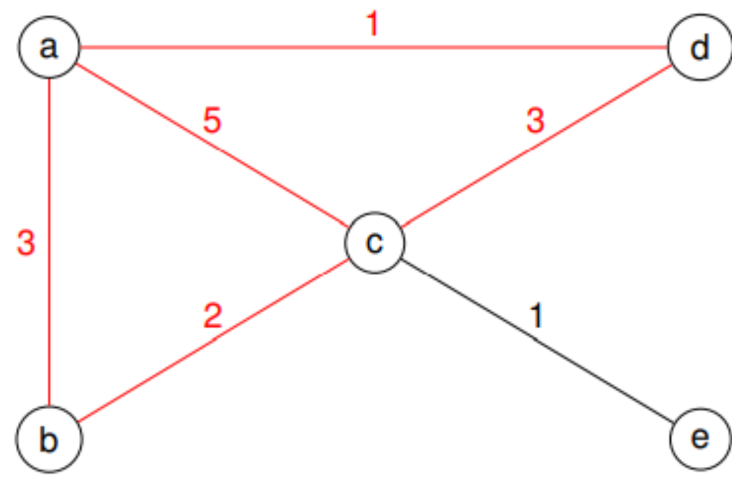
$$s.t. \quad x(a,b) + x(a,c) + x(a,d) + x(b,c) + x(c,d) + x(c,e) = 4$$

$$x(a,b) + x(a,c) + x(b,c) \leq 2$$

$$x(a,c) + x(a,d) + x(c,d) \leq 2$$

$$x(a,b) + x(a,c) + x(a,d) + x(b,c) + x(c,d) \leq 3$$

$$x(i,j) \in \{0,1\}$$



Adjazenz- und Inzidenzmatrix: Das Bus- und Bahnnetz Tegel

Erstellen Sie die zugehörige Adjazenzmatrix und Inzidenzmatrix!

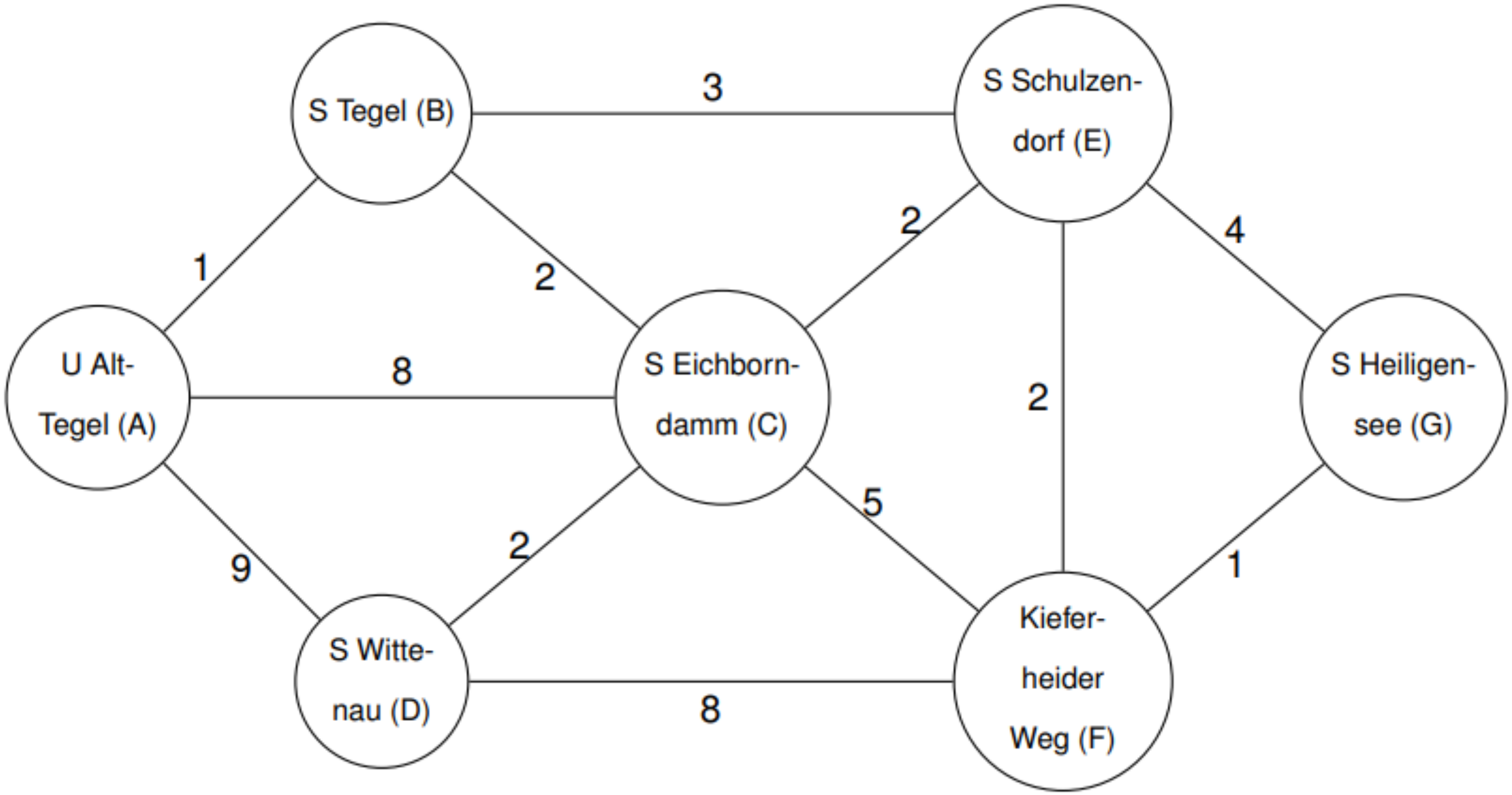


Abbildung: Verbindungen um Tegel

Adjazenz- und Inzidenzmatrix: Das Bus- und Bahnnetz Tegel

	a	b	c	d	e	f	g
a	0	1	1	1	0	0	0
b	1	0	1	0	1	0	0
c	1	1	0	1	1	1	0
d	1	0	1	0	0	1	0
e	0	1	1	0	0	1	1
f	0	0	1	1	1	0	1
g	0	0	0	0	1	1	0

Tabelle: Adjazenzmatrix

Adjazenz- und Inzidenzmatrix: Das Bus- und Bahnnetz Tegel

	e1	e2	e3	e4	e5	e6	e7	e8	e9	e10	e11	e12
a	1	1	1	0	0	0	0	0	0	0	0	0
b	1	0	0	1	0	1	0	0	0	0	0	0
c	0	1	0	1	1	0	1	1	0	0	0	0
d	0	0	1	0	1	0	0	0	1	0	0	0
e	0	0	0	0	0	1	1	0	0	1	1	0
f	0	0	0	0	0	0	0	1	1	1	0	1
g	0	0	0	0	0	0	0	0	0	0	1	1

Tabelle: Inzidenzmatrix

每个单元格的值表示节点和边之间的关系：

- 如果一个单元格的值为 1，这意味着对应的节点是由该边连接的。
- 如果一个单元格的值为 0，这意味着对应的节点不是由该边连接的。

从这个矩阵中，我们可以看到哪些站点通过特定的路线相连。例如，e1 连接着节点 a 和 b，因为在e1列中，a 和 b 行的值为 1。类似地，e3 连接着节点 b 和 c。

在图论中，Inzidenzmatrix（即邻接矩阵）是用于表示图中节点与边之间关系的矩阵。对于无向图，邻接矩阵的元素通常是0或1，其中1表示两个节点之间存在边，0表示没有边。但是对于有向图，我们通常使用-1，0和1来表示节点与边的关系：

- `0` 表示节点和边没有关联。
- `1` 表示边的起点是该节点。
- `-1` 表示边的终点是该节点。

Travelling Salesman Problem – Einführendes Beispiel

Kürzeste Rundtouren

Willkommen zum TSP-Spiel!

Auf dieser Seite geht es um das sogenannte "Problem des Handlungsreisenden" oder auch "Traveling Salesman Problem".

Dieses Problem beschäftigt sich mit der Frage, wie man eine Tour durch eine bestimmte Anzahl Städte und zurück zum Ausgangspunkt planen muss, damit der insgesamt zurückgelegte Weg möglichst klein ist. Sie ist beispielsweise für vier Städte sehr leicht zu beantworten, doch bereits bei zehn Städten sind potenziell 181.440 verschiedene solcher Rundtouren möglich.

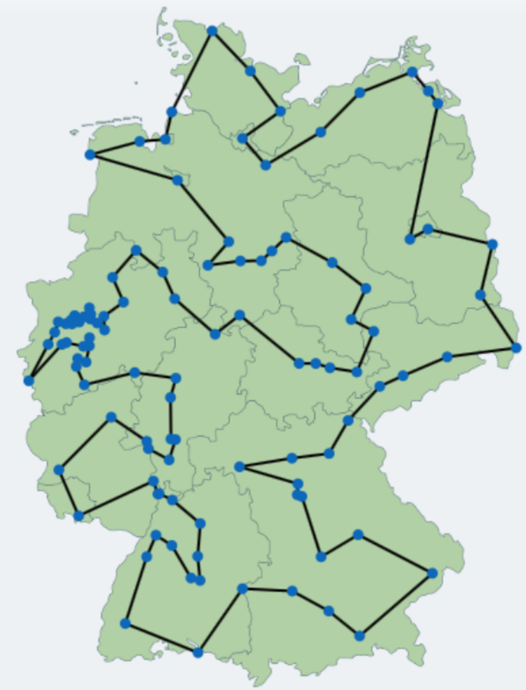
Wie schafft man es also, zwanzig oder noch mehr Städte auf dem kürzesten Weg zu besuchen?

Zu diesem Zweck wurden in den letzten Jahrzenten verschiedene Algorithmen entwickelt, die das Problem des Handlungsreisenden möglichst schnell beantworten.

Doch jetzt bist du am Zug!

Erstelle im nächsten Tab ein Spiel und versuche selbst, die kürzeste Rundtour zu finden! Anschließend kannst du dir die Lösung, sowie die Berechnungsschritte der verwendeten Algorithmen ansehen.

Viel Spaß!



Was möchtest du zuerst tun?

Ein neues Spiel erstellen! | Genauere Beschreibung des Algorithmus lesen!

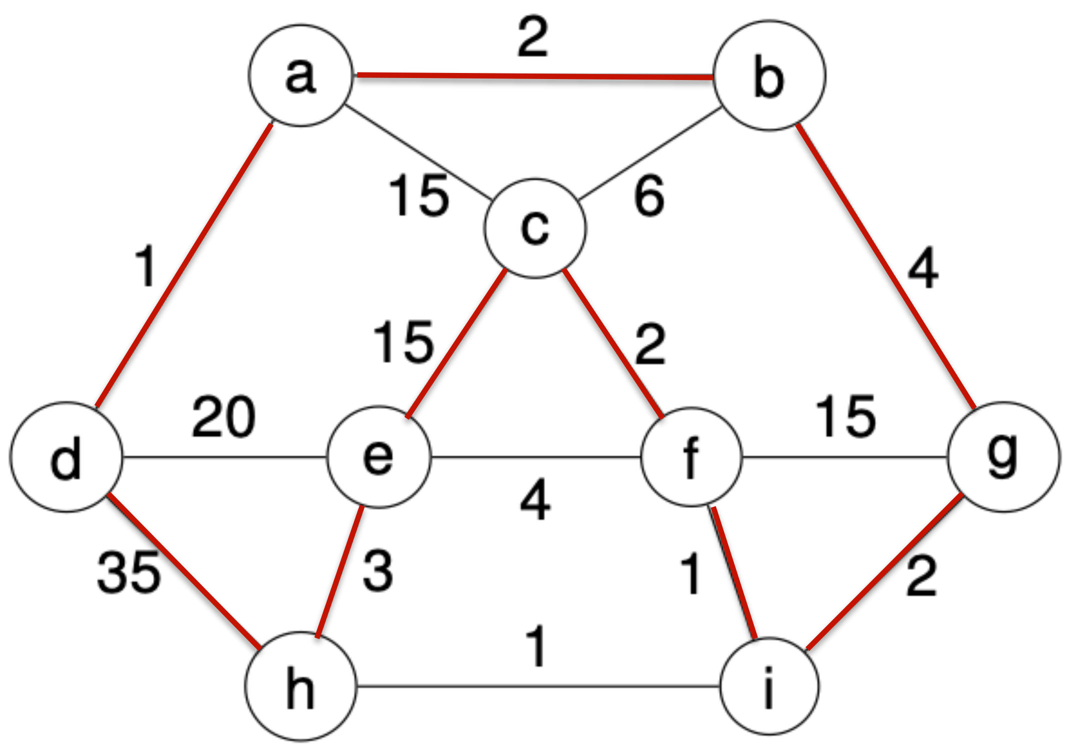
Das TSP-Spiel der Technischen Universität München

https://www-m9.ma.tum.de/games/tsp-game/index_de.html

Travelling Salesman Problem

Was berechnet uns das Travelling Salesman Problem (TSP)?

- Den kürzesten geschlossenen Weg, in dem jeder Knoten genau einmal enthalten ist



Travelling Salesman Problem

Allgemeine mathematische Formulierung

V: Knoten

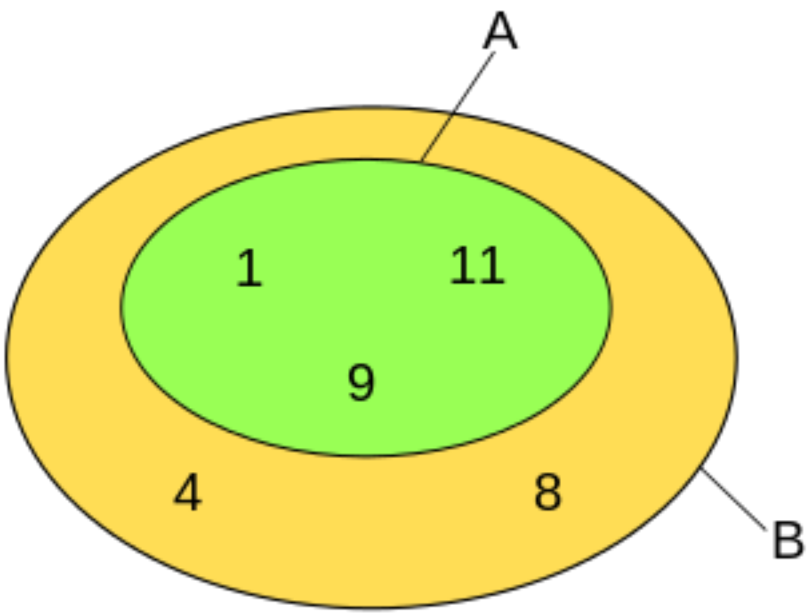
i,j: Beliebige Knoten

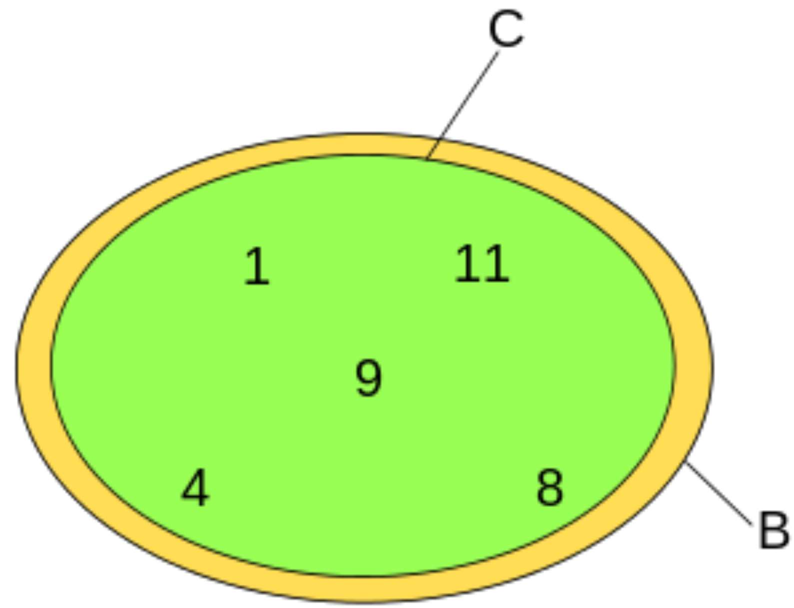
n: Anzahl Knoten im Graphen

S: Jede <u>echte</u> Teilmenge von V

ZF:	$\min z = \sum_{(i,j)\in E} w(i,j)\cdot x(i,j)$				
NB1:	s.t. $\sum_{j=1}^{n}$	$x(i,j)$	$=$	1	für $i = 1,\ldots,n$
NB2:	$\sum_{i=1}^{n}$	$x(i,j)$	$=$	1	für $j = 1,\ldots,n$
NB3:	$\sum_{(i,j)\in E(S)}$	$x(i,j)$	$\leq$	$\lvert S\rvert - 1$	für jede Teilmenge $S \subset V$
NBX:		$w(i,j)$	$\in$	$\mathbb{R}$	
Def.B.:		$x(i,j)$	$\in$	$\{0,1\}$	

Echte vs. Unechte Teilmengen





$A \subset B$

A ist eine echte Teilmenge von B

$A \subseteq B$

C ist keine echte Teilmenge von B

Travelling Salesman Problem

Allgemeine mathematische Formulierung

V: Knoten

i,j: Beliebige Knoten

n: Anzahl Knoten im Graphen

S: Jede echte Teilmenge von V

E: Kanten; E(S): Alle Kanten deren beiden Endpunkte in S liegen; |S|: Anzahl der Knoten in S

w(i,j): Kantengewicht der Kante zwischen den Knoten i und j

x(i,j): Binäre Entscheidungsvariable der Kante zwischen i und j

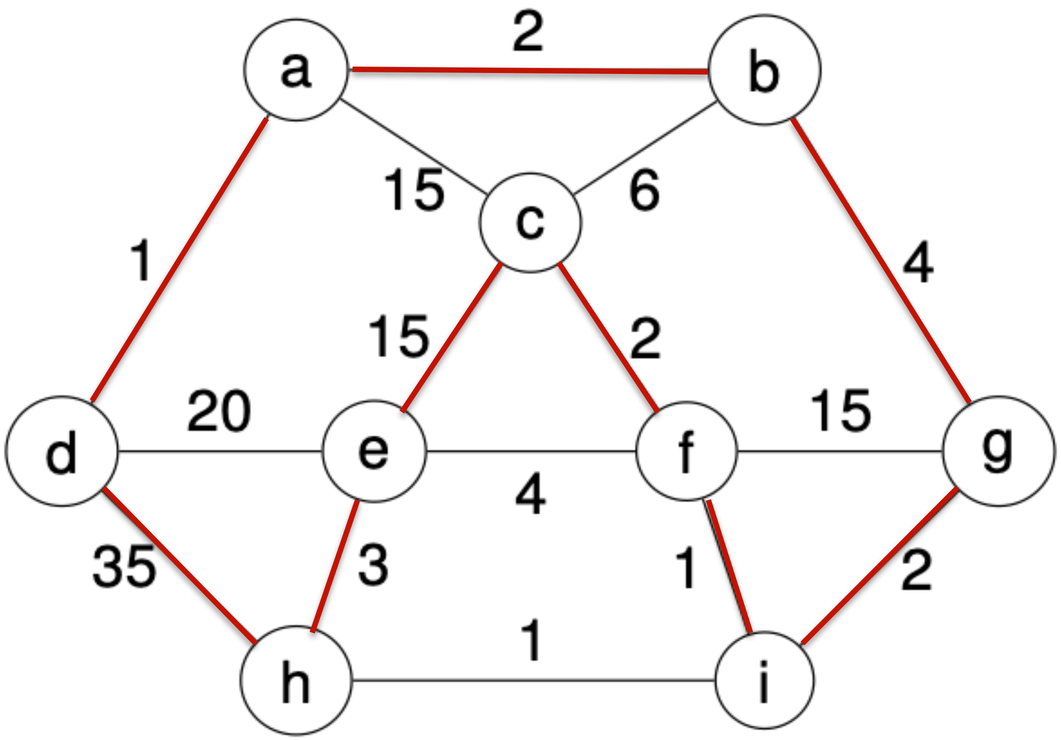
(x(i,j) = 0 → Kante ist nicht in der Lösung enthalten / x(i,j) = 1 → Kante ist in der Lösung enthalten)

(Achtung für jede Kante existieren zwei x-Variablen: eine für jede Richtung!)

ZF:	$\min z =$	$\sum_{(i,j)\in E}$	$w(i,j)\cdot x(i,j)$			
NB1:	s.t.	$\sum_{j=1}^{n}$	$x(i,j)$	$=$	1	für $i = 1, \ldots, n$
NB2:		$\sum_{i=1}^{n}$	$x(i,j)$	$=$	1	für $j = 1, \ldots, n$
NB3:		$\sum_{(i,j)\in E(S)}$	$x(i,j)$	$\leq$	$\lvert S\rvert - 1$	für jede Teilmenge $S \subset V$
NBX:			$w(i,j)$	$\in$	$\mathbb{R}$	
Def.B.:			$x(i,j)$	$\in$	$\{0,1\}$	

Travelling Salesman Problem

Bsp. Für Funktionsweise von x(i,j)



$$x(a,b)^* = 1 \qquad x(a,c)^* = 0$$
$$x(b,g)^* = 1 \qquad x(b,c)^* = 0$$
$$x(g,i)^* = 1 \qquad x(d,e)^* = 0$$
$$\ldots \qquad \ldots$$

Travelling Salesman Problem

目标函数 (ZF): 最小化整个路径的总权重。即对图中所有边的权重与对应决策变量的乘积求和，决策变量表示相应的边是否被选择。
约束条件 (NB1): 对于每个节点，流出的边的总数必须等于1。这意味着从每个节点只能有一条边流出。
约束条件 (NB2): 对于每个节点，流入的边的总数也必须等于1。这意味着到每个节点只能有一条边流入。
约束条件 (NB3): 对于图中的任何节点子集 S，从 S 出发或到达 S 的边的总数必须小于或等于 S 中节点的数量减1。这个约束通常用于防止在子图中形成闭环。
额外约束 (NBX): 边的权重是实数。
决策变量的定义 (Def. B.): 决策变量 $x(i,j)$ 是二元的，即可以取0或1的值。

Umgangssprachliche Formulierung

ZF:	$\min z =$	$\sum_{(i,j)\in E}$	$w(i,j)\cdot x(i,j)$			
NB1:	s.t.	$\sum_{j=1}^{n}$	$x(i,j)$	$=1$		für $i=1,\ldots,n$
NB2:		$\sum_{i=1}^{n}$	$x(i,j)$	$=1$		für $j=1,\ldots,n$
NB3:		$\sum_{(i,j)\in E(S)}$	$x(i,j)$	$\leq$	$\lvert S\rvert-1$	für jede Teilmenge $S\subset V$
NBX:			$w(i,j)$	$\in$	$\mathbb{R}$	
Def. B.:			$x(i,j)$	$\in$	$\{0,1\}$	

ZF: Minimiere die gesamte Weglänge der Lösung: Summiere das Produkt aus Kantengewicht und Entscheidungsvariable für jede Kante auf

NB1: Für jeden Knoten muss gelten: Die Summe der rausführenden Kanten = 1

NB2: Für jeden Knoten muss gelten: Die Summe der reinführenden Kanten = 1

NB3: 1) Für jede Kante muss gelten: Die Summe der beiden x-Variablen (eine für jede Richtung) $\leq$ 1

2) Für jeden Kreis, der nicht alle Knoten enthält, muss gelten: Die Summe der Kanten des Kreises $\leq$ Anzahl der Kanten des Kreises - 1

Travelling Salesman Problem

Umgangssprachliche Formulierung

ZF:	$\min z =$	$\sum_{(i,j)\in E}$	$w(i,j) \cdot x(i,j)$			
NB1:	s.t.	$\sum_{j=1}^{n}$	$x(i,j)$	$= 1$		für $i = 1, \ldots, n$
NB2:		$\sum_{i=1}^{n}$	$x(i,j)$	$= 1$		für $j = 1, \ldots, n$
NB3:		$\sum_{(i,j)\in E(S)}$	$x(i,j)$	$\leq$	$\lvert S\rvert - 1$	für jede Teilmenge $S \subset V$
NBX:			$w(i,j)$	$\in$	$\mathbb{R}$	
Def. B.:			$x(i,j)$	$\in$	$\{0,1\}$	

NBX: Diese NB muss bei der Anwendung des TSP nicht aufgestellt werden, da die Kantengewichte nicht als w(i,j) angegeben werden, sondern konkrete Zahlenwerte verwendet werden.

Def. B: Die Entscheidungsvariable jeder Kante ist binär und hat daher den Definitionsbereich {0,1}

2.34 Travelling Salesman Problem

Ein dem Travelling Salesman Problem (TSP) zugrunde liegendes praktisches Problem lässt sich wie folgt schildern: Ein Handlungsreisender möchte eine Rundreise durch verschiedene Städte planen. Dabei möchte er in seinem Wohnort startend und am Ende auch dorthin zurückkehrend. Die Aufgabe besteht nun darin die Reihenfolge, in welcher die n Städte aufgesucht werden, so zu bestimmen, dass die zurückgelegte Gesamtstrecke minimal wird.

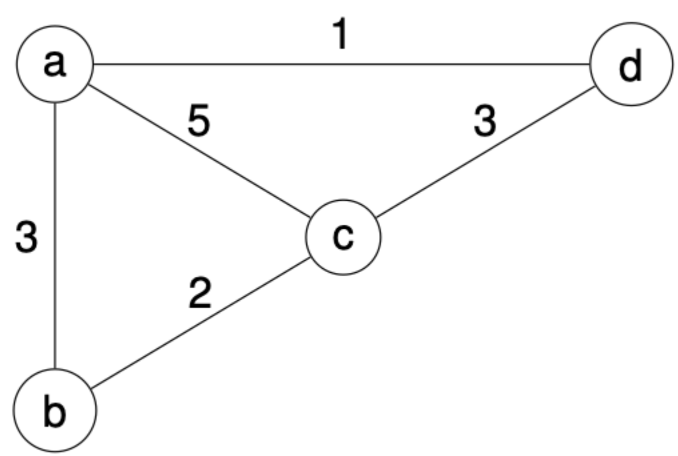
a) Stellen Sie das TSP allgemein als Optimierungsproblem dar.

ZF:	$\min z =$	$\sum_{(i,j)\in E}$	$w(i,j)\cdot x(i,j)$			
NB1:	s.t.	$\sum_{j=1}^{n}$	$x(i,j)$	$=1$		für $i=1,\ldots,n$
NB2:		$\sum_{i=1}^{n}$	$x(i,j)$	$=1$		für $j=1,\ldots,n$
NB3:		$\sum_{(i,j)\in E(S)}$	$x(i,j)$	$\leq$	$\|S\|-1$	für jede Teilmenge $S\subset V$
NBX:			$w(i,j)$	$\in$	$\mathbb{R}$	
Def. B.:			$x(i,j)$	$\in$	$\{0,1\}$	

2.34 Travelling Salesman Problem

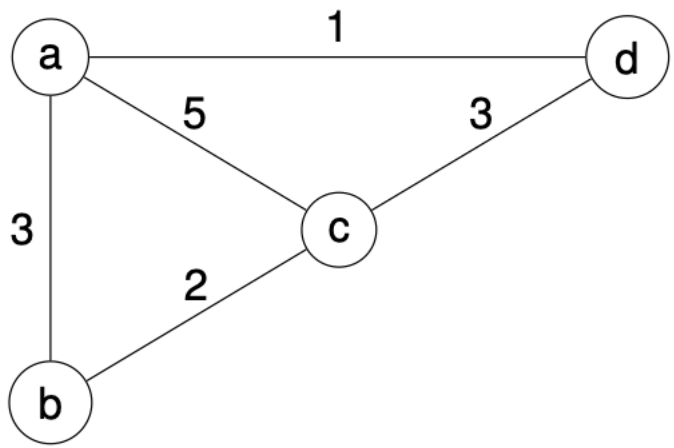
Ein dem Travelling Salesman Problem (TSP) zugrunde liegendes praktisches Problem lässt sich wie folgt schildern: Ein Handlungsreisender möchte eine Rundreise durch verschiedene Städte planen. Dabei möchte er in seinem Wohnort startend und am Ende auch dorthin zurückkehrend. Die Aufgabe besteht nun darin die Reihenfolge, in welcher die n Städte aufgesucht werden, so zu bestimmen, dass die zurückgelegte Gesamtstrecke minimal wird.

b) Nun stellen sie das spezielle Problem für den gegebenen Graphen auf.



2.34 Travelling Salesman Problem

b) Nun stellen sie das spezielle Problem für den gegebenen Graphen auf.



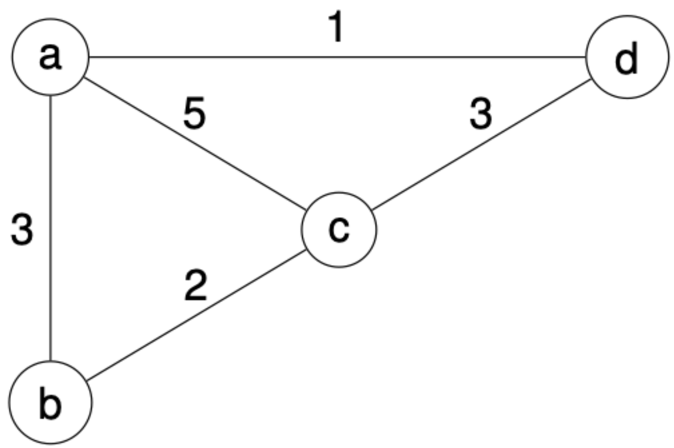
ZF: $\min z = \sum_{(i,j)\in E} w(i,j) \cdot x(i,j)$

Minimiere die gesamte Weglänge der Lösung: Summiere das Produkt aus Kantengewicht und Entscheidungsvariable für jede Kante auf

$$\min z = 3 * x(a,b) + 5 * x(a,c) + 1 * x(a,d) + 2 * x(b,c) + 3 * x(c,d)$$
$$+3 * x(b,a) + 5 * x(c,a) + 1 * x(d,a) + 2 * x(c,b) + 3 * x(d,c)$$

2.34 Travelling Salesman Problem

b) Nun stellen sie das spezielle Problem für den gegebenen Graphen auf.



NB1: $\sum_{j=1}^{n} \quad x(i,j) \quad = 1 \qquad$ für $i = 1, \dots, n$

Für jeden Knoten muss gelten: Die Summe der rausführenden Kanten = 1

Knoten a: $x(a,b) + x(a,c) + x(a,d) = 1$

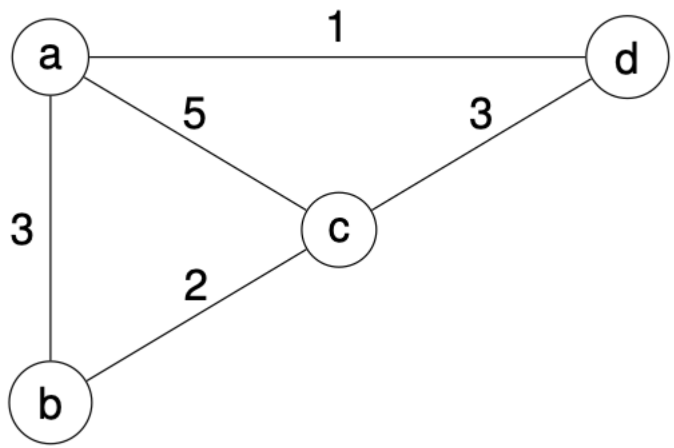
Knoten b: $x(b,a) + x(b,c) = 1$

Knoten c: $x(c,a) + x(c,b) + x(c,d) = 1$

Knoten d: $x(d,a) + x(d,c) = 1$

2.34 Travelling Salesman Problem

b) Nun stellen sie das spezielle Problem für den gegebenen Graphen auf.



NB2: $\sum_{i=1}^{n} \quad x(i,j) \quad = 1$ für $j = 1, \ldots, n$

Für jeden Knoten muss gelten: Die Summe der reinführenden Kanten = 1

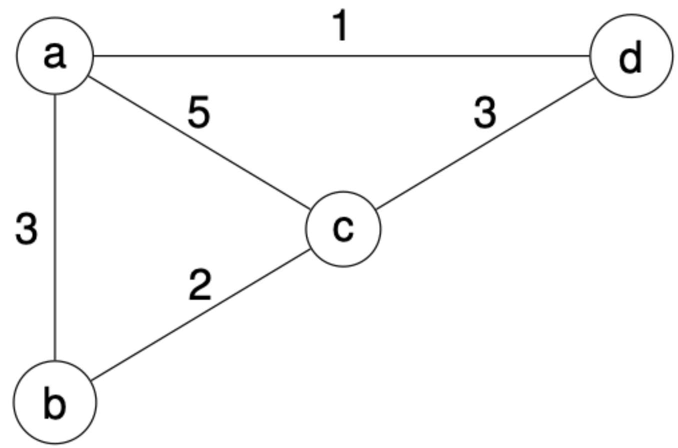
Knoten a: $x(b,a) + x(c,a) + x(d,a) = 1$

Knoten b: $x(a,b) + x(c,b) = 1$

Knoten c: $x(a,c) + x(b,c) + x(d,c) = 1$

Knoten d: $x(a,d) + x(c,d) = 1$

2.34 Travelling Salesman Problem



NB3: $\sum_{(i,j)\in E(S)} x(i,j) \leq |S|-1$ für jede Teilmenge $S \subset V$

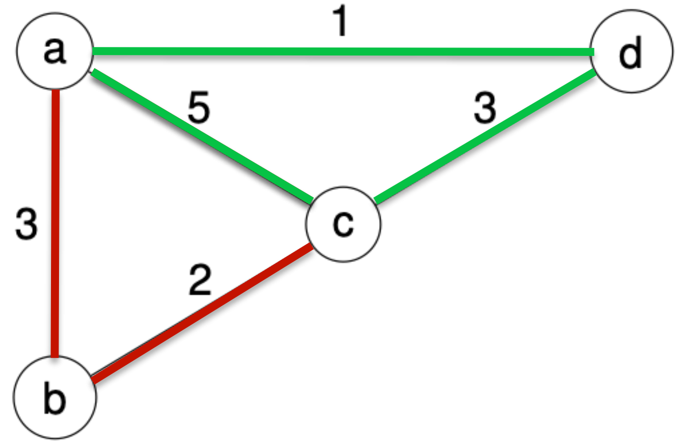
1) Für jede Kante muss gelten: Die Summe der beiden x-Variablen (eine für jede Richtung) ≤ 1

2) Für jeden Kreis, <u>der nicht alle Knoten enthält</u>, muss gelten: Die Summe der Kanten des Kreises ≤ Anzahl der Kanten des Kreises – 1

1)

$$x(a,b)+x(b,a)\leq 1$$
$$x(a,c)+x(c,a)\leq 1$$
$$x(a,d)+x(d,a)\leq 1$$
$$x(b,c)+x(c,b)\leq 1$$
$$x(c,d)+x(d,c)\leq 1$$

2.34 Travelling Salesman Problem



NB3: $\sum_{(i,j)\in E(S)} \quad x(i,j) \quad \leq \quad |S|-1$ für jede Teilmenge $S \subset V$

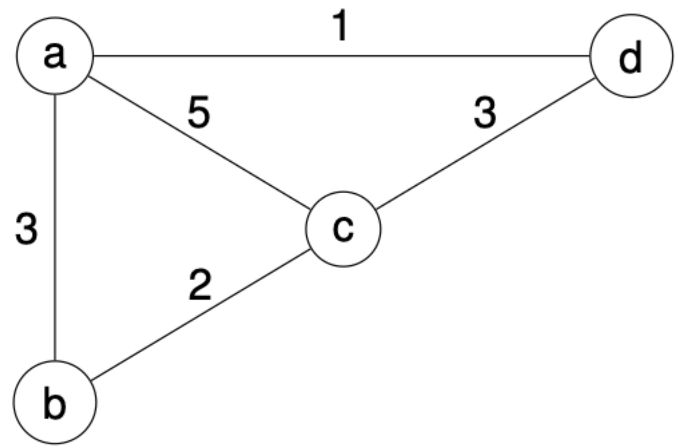
1) Für jede Kante muss gelten: Die Summe der beiden x-Variablen (eine für jede Richtung) ≤ 1

2) Für jeden Kreis, <u>der nicht alle Knoten enthält</u>, muss gelten: Die Summe der Kanten des Kreises ≤ Anzahl der Kanten des Kreises – 1

2)

$$x(a,b) + x(b,c) + x(c,a) \leq 2$$
$$x(a,c) + x(c,b) + x(b,a) \leq 2$$
$$x(a,c) + x(c,d) + x(d,a) \leq 2$$
$$x(a,d) + x(d,c) + x(c,a) \leq 2$$

2.34 Travelling Salesman Problem



Def. B. : $x(i,j) \quad \in \quad \{0,1\}$

Die Entscheidungsvariable jeder Kante ist binär und hat den Definitionsbereich {0,1}

$$x(i,j) \in \{0,1\} \quad \forall\, i,j \in \{a,b,c,d\}$$

2.34 Travelling Salesman Problem

Gesamte Lösung

$$
\begin{array}{llllllr}
\min z = & 3\cdot x(a,b) & +5\cdot x(a,c) & +1\cdot x(a,d) & +2\cdot x(b,c) & +3\cdot x(c,d) & \\
 & +3\cdot x(b,a) & +5\cdot x(c,a) & +1\cdot x(d,a) & +2\cdot x(c,b) & +3\cdot x(d,c) & \\
\text{s.t.} & x(a,b) & +x(a,c) & +x(a,d) & & & =1 \\
 & x(b,c) & +x(b,a) & & & & =1 \\
 & x(c,a) & +x(c,b) & +x(c,d) & & & =1 \\
 & x(d,c) & +x(d,a) & & & & =1 \\
 & x(b,a) & +x(c,a) & +x(d,a) & & & =1 \\
 & x(c,b) & +x(a,b) & & & & =1 \\
 & x(a,c) & +x(b,c) & +x(d,c) & & & =1 \\
 & x(c,d) & +x(a,d) & & & & =1 \\
 & x(c,d) & +x(d,c) & & & & \leq 1 \\
 & x(d,a) & +x(a,d) & & & & \leq 1 \\
 & x(c,a) & +x(a,c) & & & & \leq 1 \\
 & x(b,a) & +x(a,b) & & & & \leq 1 \\
 & x(c,b) & +x(b,c) & & & & \leq 1 \\
 & x(a,b) & +x(b,c) & +x(c,a) & & & \leq 2 \\
 & x(a,c) & +x(c,d) & +x(d,a) & & & \leq 2 \\
 & x(b,a) & +x(a,c) & +x(c,b) & & & \leq 2 \\
 & x(c,a) & +x(a,d) & +x(d,c) & & & \leq 2 \\
 & & & & x(i,j) \in & \{0,1\} &
\end{array}
$$

Achtung!! Hier ist der Definitionsbereich falsch angegeben!
Richtig: $x(i,j) \in \{0,1\} \quad \forall\, i,j \in \{a,b,c,d\}$

Fragen zum Tutorium?

Fragen zum Tutorium?