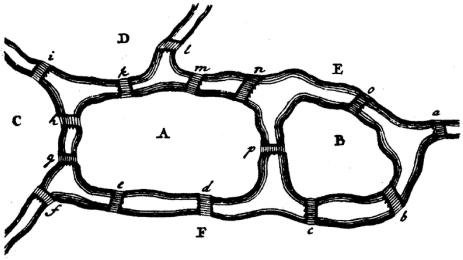
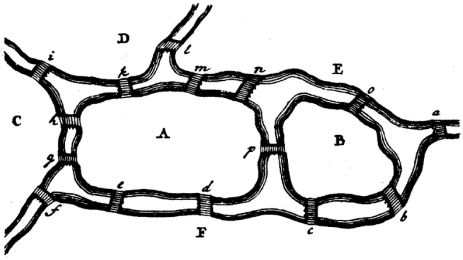
Agenda

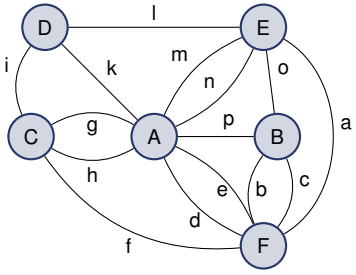
Agenda

Königsberger Brückenproblem (Euler, 1736)

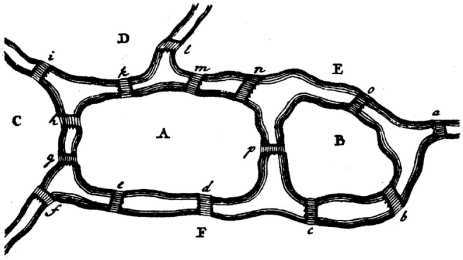


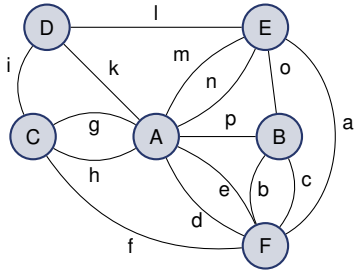
Königsberger Brückenproblem (Euler, 1736)





Königsberger Brückenproblem (Euler, 1736)



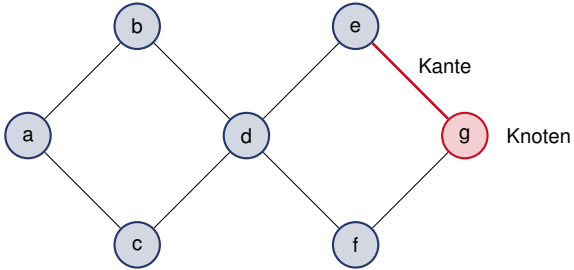


Definition

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

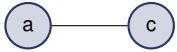
Definition

Ein **Graph** $g(V,E)$ ist eine Menge von Punkten V (**Ecken**/**Knoten**, *vertex/node*), die eventuell durch **Kanten** E (*edge*) miteinander verbunden sind, wobei es nicht auf die Form ankommt.

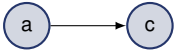


Definitionen

Ungerichtete Kante (*undirected edge*)



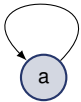
Gerichtete Kante (*directed edge*)



Parallele Kanten

(wenn gerichtet: in die gleiche Richtung)

Schlinge (*loop*)



Endlicher Graph, schlichter Graph, Multigraph

Definition

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

Definition

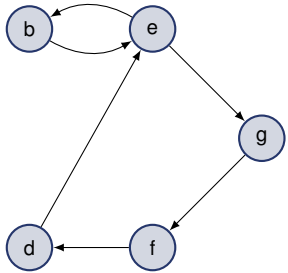
Ein Graph ohne parallele Kanten und ohne Schlinge wird als **schlichter Graph** bezeichnet.

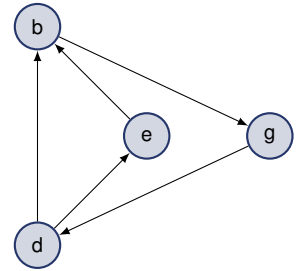
Definition

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

Definition

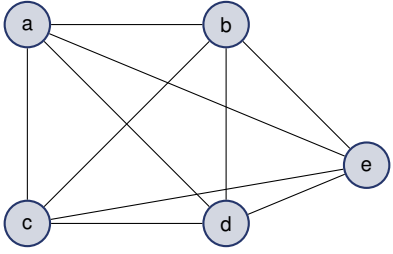
Ein schlichter, gerichteter Graph mit endlicher Knotenmenge heißt **Digraph**.





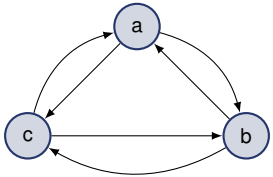
Definition

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



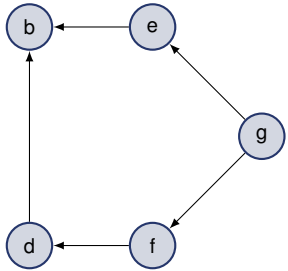
Definition

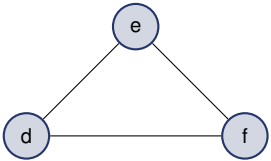
Ein Digraph heißt **vollständiger Digraph**, wenn für jedes Knotenpaar i, j ein Pfeil von i nach j und ein Pfeil von j nach i vorhanden ist.



Definition

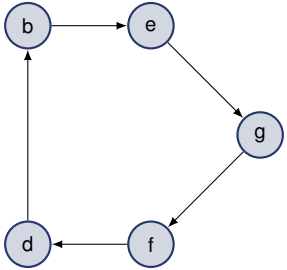
Eine **Kette** ist die Folge von ungerichteten oder gerichteten Kanten, wobei der Richtungssinn keine Rolle spielt. Eine geschlossene Kette heißt **Kreis**.





Definition

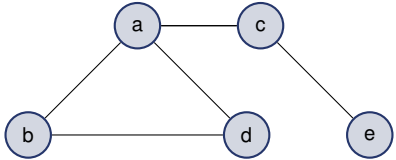
Ein **Weg** ist die Folge von gerichteten Kanten, jeweils vom Pfeilende zur Pfeilspitze. Ein geschlossener Weg heißt **Zyklus**.



Zusammenhängender Graph

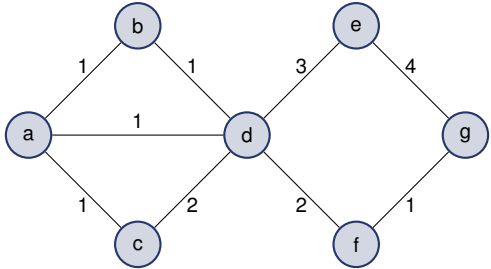
Definition

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



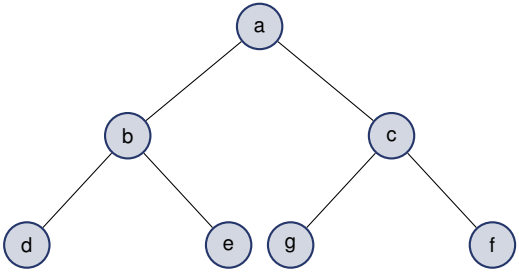
Definition

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 Pfeilbewertungen eines Weges wird auch als Länge des Weges bezeichnet.

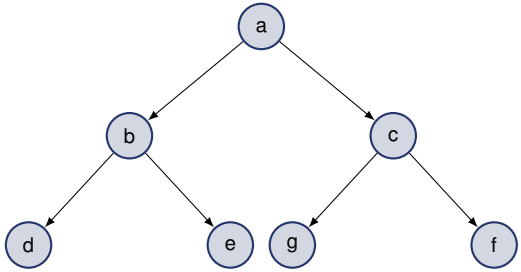


Definition

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



ungerichterer Baum



gerichterer Baum

Definition

In einem gerichteten Graphen heißt ein Knoten *j* **Nachfolger** eines Knoten *i*, wenn ein Weg vom Knoten *i* zum Knoten *j* existiert. Umgekehrt ist *i* ein **Vorgänger** von *j*. Vorgänger und Nachfolger werden auch als **Nachbarn** bezeichnet. In ungerichteten Graphen spricht man nur von Nachbarn. Sollten Knoten *j* und *i* nur durch eine Kante verbunden sein, spricht man von unmittelbaren Nachbarn.

- Ein Knoten ohne Vorgänger heißt **Quelle** (*source*)
- Ein Knoten ohne Nachfolger heißt **Senke** (*sink*)

在一个有向图中，如果存在从节点 i 到节点 j 的路径，则节点 j 被称为节点 i 的后继。反之，节点 i 是节点 j 的前驱。前驱和后继也被称为邻居。在无向图中，只谈论邻居。如果节点 j 和节点 i 仅由一条边连接，则称它们是直接相邻的。

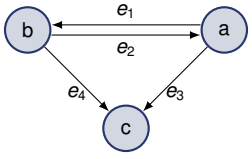
没有前驱的节点称为源（source）
没有后继的节点称为汇（sink）

Definition

Die **Adjazenzmatrix** $A(g) = a_{ij}$ drückt aus, welche Knoten i und j im Graph g verbunden sind. [n x n]-Matrix mit binären Elementen (0;1):

- 1 wenn $e(i,j)$ existiert
- 0 in allen anderen Fällen

Für ungerichtete Graphen ist die Adjazenzmatrix symmetrisch.
adjazent = benachbart

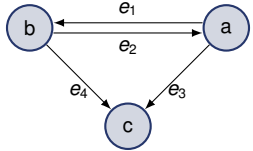


Definition

Die **Inzidenzmatrix** $H(g) = h_{ij}$ drückt aus, welche Kanten e_j mit Knoten i im Graph g verbunden sind. [n x m]-Matrix mit folgenden Elementen:

- 1 wenn i der Startpunkt von e_j ist
- -1 wenn i der Endpunkt von e_j ist (nur in gerichteten Graphen)
- 0 in allen anderen Fällen (wenn i weder Start- noch Endpunkt von e_j ist).

inzident $\approx$ *verbunden*



Agenda

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, und damit
- (V',E') ist zusammenhängend.

生成树
定义
给定一个带有边权重的无向图 g(V, E, w)。一个(V, E, w)的生成树是(V, E)的一个子图，满足以下条件：
- V′ = V，即子图包含 g 的所有节点，
- (V′, E′) 是一棵树，因此
- (V′, E′) 是连通的。

Definition

Das **Gewicht** $l(V',E')$ eines Spannbaumes ist die Summe aller seiner Kantengewichte: $l(V',E') = \sum_{e \in E'} w(e)$

定义
生成树的权重 l(V′, E′) 是其所有边权重的总和：l(V′, E′) = Σ

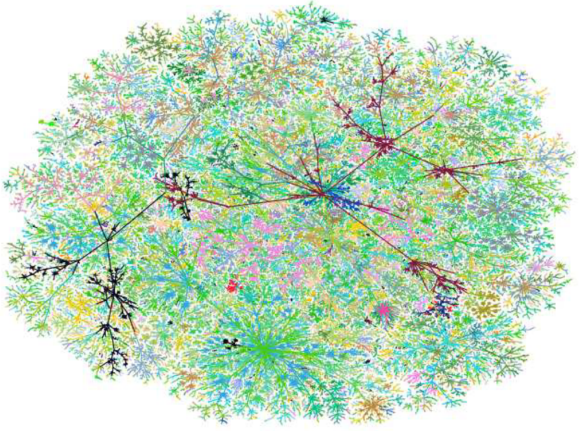
Definition

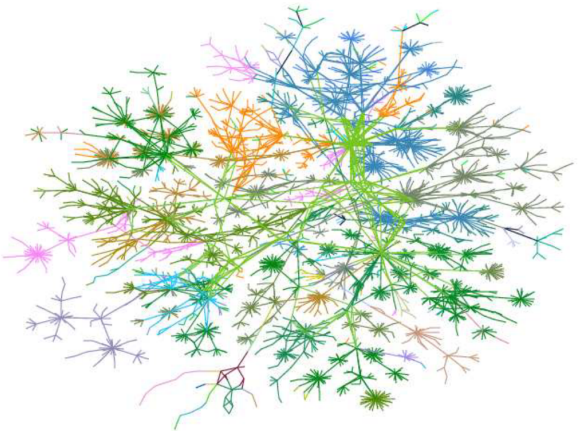
Ein **minimaler Spannbaum** hat minimales Gewicht unter allen möglichen Spannbäumen.

定义
最小生成树是所有可能的生成树中具有最小权重的生成树。

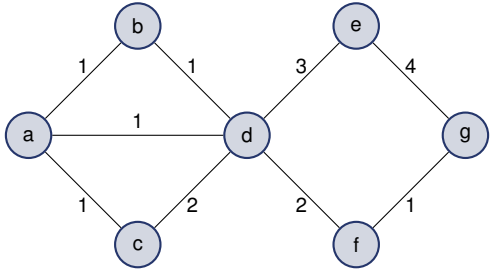
Anwendungsbeispiele:

- Linienplanung für den öffentlichen Personenverkehr
- Kostengünstigstes Anschlusskabel für Internet

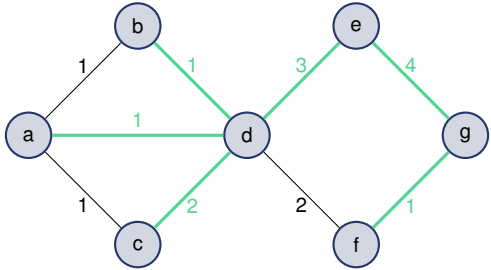




Spannbäume – Beispiel

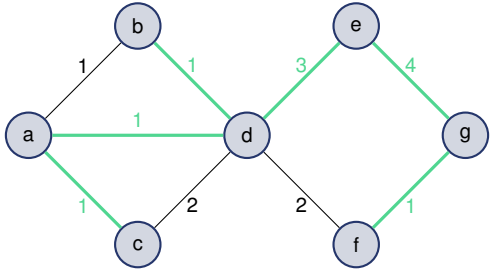


Spannbäume – Beispiel



Gewicht: 12

Spannbäume – Beispiel



Gewicht: 11

Joseph B. Kruskal

Leben:

- *29.01.1928 (New York City) †19.09.2010 Princeton
- US-amerikanischer Mathematiker und Statistiker
- Studium an der Universität von Chicago und Princeton Universität
- 1954: Promotion an der Princeton Universität, New Jersey
- arbeitete als Assistenzprofessor an der „University of Michigan“ sowie für das Unternehmen „Bell Laboratories“

Hauptwerk:

- Satz über die Ordnungseigenschaft einer unendlichen Folge endlicher Bäume (1960)

Wirkung:

- Kruskal-Algorithmus

Kruskal-Algorithmus

Bedingungen

- Graph muss
  - ▷ endlich,
  - ▷ zusammenhängend,
  - ▷ schlingenfrei,
  - ▷ ungerichtet
  - ▷ und gewichtet sein.

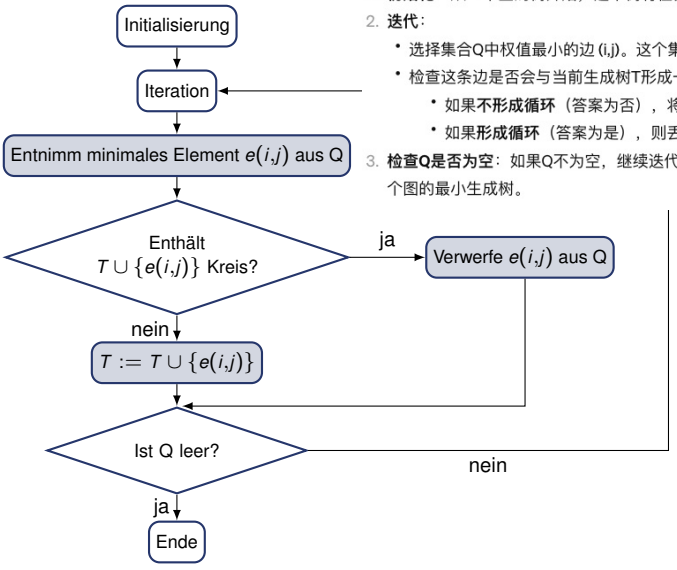
Variablen

- Liste T ausgewählter Kanten $e(i,j)$

Anfangszustand

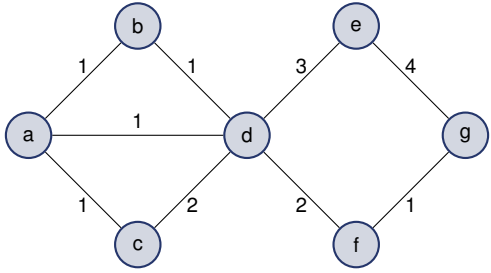
- $T = \emptyset$
- Eine Liste Q mit allen der Länge nach aufsteigend sortierten Kanten

Kruskal-Algorithmus

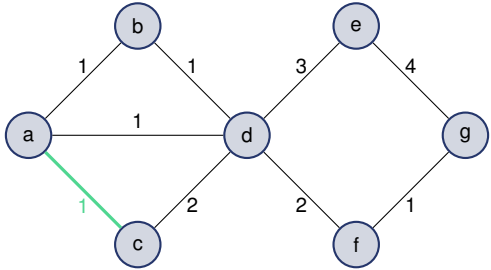


1. **初始化**：从一个空的树开始，这个树将在算法的过程中逐步建立。
2. **迭代**：
   - 选择集合Q中权值最小的边 (i,j)。这个集合Q包含了图中所有的边。
   - 检查这条边是否会与当前生成树T形成一个循环：
     - 如果**不形成循环**（答案为否），将边 (i,j) 加入到T中。
     - 如果**形成循环**（答案为是），则丢弃这条边，不将其加入T。
3. **检查Q是否为空**：如果Q不为空，继续迭代；如果Q为空，算法结束，此时T就是该个图的最小生成树。

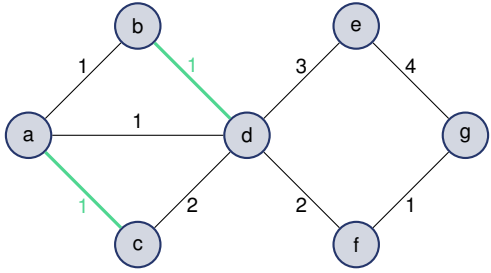
Kruskal-Algorithmus



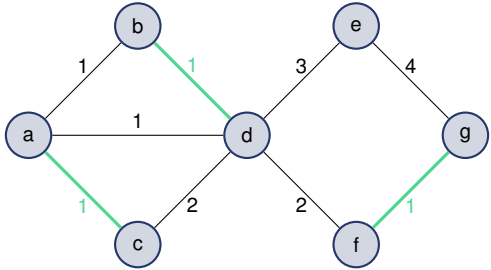
Kruskal-Algorithmus



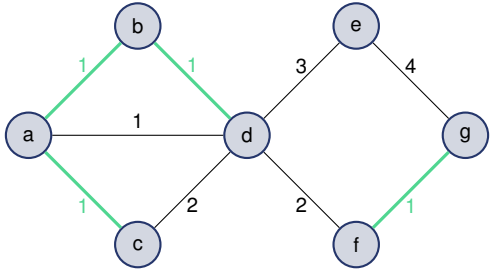
Kruskal-Algorithmus



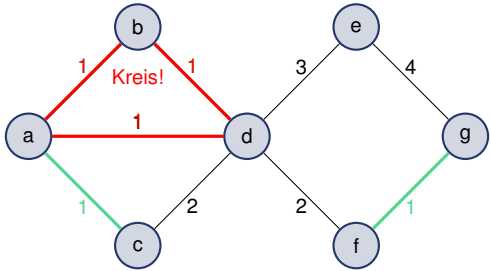
Kruskal-Algorithmus



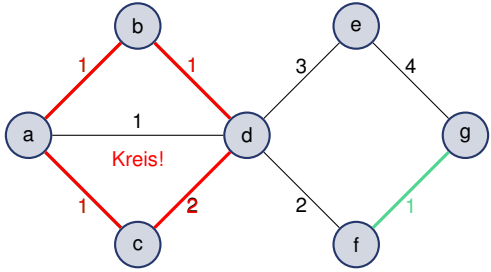
Kruskal-Algorithmus



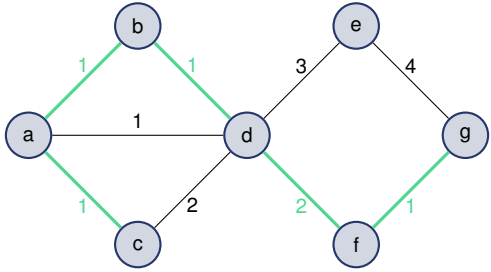
Kruskal-Algorithmus



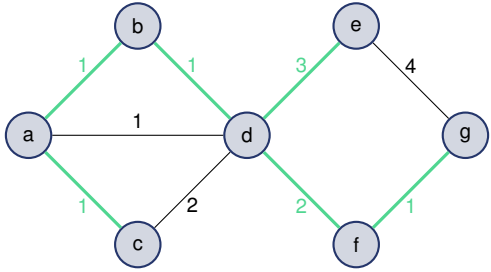
Kruskal-Algorithmus



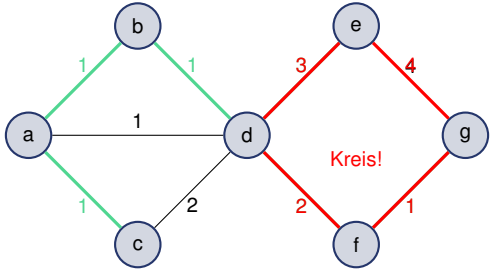
Kruskal-Algorithmus



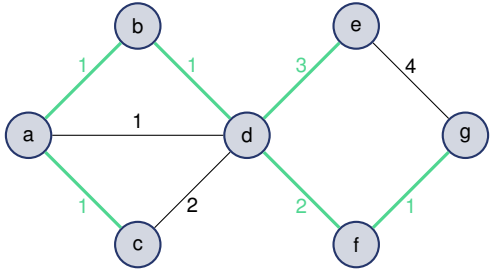
Kruskal-Algorithmus



Kruskal-Algorithmus



Kruskal-Algorithmus



Minimaler Spannbaum, Gewicht: 9

Minimale Spannbäume als Optimierungsproblem

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. Daher gilt: Die Lösung des folgenden Optimierungsproblems beschreibt einen minimalen Spannbaum für $g(V, E, w)$ mit $w(i,j) \in \mathbb{R}$ als Parameter für die Kantengewichte.

$$
\begin{aligned}
\min z = \quad & \sum_{(i,j)\in E} && w(i,j) \cdot x_{i,j} && && \\
\text{s.t.} \quad & \sum_{(i,j)\in E} && x(i,j) && = && n-1 \\
& \sum_{j:(s,j)\in E(S)} && x(i,j) && \leq && |S|-1 \quad \text{für jede Teilmenge } S \subset V \text{ von Knoten.} \\
& && x(i,j) && \leq && 1 \qquad\qquad \text{für } (i,j) \in E \\
& && x(i,j) && \geq && 0
\end{aligned}
$$

这张图片上描述了最小生成树问题作为一个优化问题的数学表述。具体地，它定义一个图 $G(V, E, w)$，其中 V 是节点的集合，E 是边的集合，w 是边的权重函数，$w(i, j)$ 表示连接节点 i 和 j 的边的权重。优化问题旨在找到图的一个生成树，使得树的总边权重最小。

这个优化问题可以使用以下的线性规划模型来表述：

目标函数：

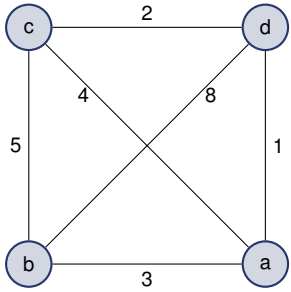
$\min z = \sum_{(i,j)\in E} w(i,j) \cdot x_{ij}$

这是所有边权重和边是否在生成树中的乘积之和，我们希望最小化这个总和。

1. $\sum_{(i,j)\in E} x_{ij} = n - 1$
   这个约束确保生成树中恰好有 $n-1$ 条边，这是连接 n 个节点的树所必须的边的数量。
2. $\sum_{(i,j)\in E(S)} x_{ij} \leq |S| - 1$ 对于所有 $S \subset V$ 的节点子集。
   这组约束防止在生成树中出现循环。对于图中的任何节点子集，连接这些节点的边的数量必须少于该子集中节点的数量。
3. $x_{ij} \leq 1$ 对于所有 $(i, j) \in E$。
   这表示一条边要么在生成树中（取值 1），要么不在（取值 0）。
4. $x_{ij} \geq 0$ 对于所有 $(i, j) \in E$。
   这是一个典型的非负约束，确保每条边的指示变量 x_{ij} 不会取负数。

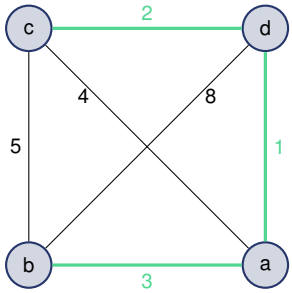
Minimale Spannbäume als Optimierungsproblem

$$
\begin{array}{lrrrrrrrl}
\min z = & 3 \cdot x_{ab} & +4 \cdot x_{ac} & +1 \cdot x_{ad} & +5 \cdot x_{bc} & +8 \cdot x_{bd} & +2 \cdot x_{cd} & & \\
\text{s.t.} & x_{ab} & +x_{ac} & +x_{ad} & +x_{bc} & +x_{bd} & +x_{cd} & = & 3 \\
 & x_{ab} & & & & & & \leq & 1 \\
 & & x_{ac} & & & & & \leq & 1 \\
 & & & x_{ad} & & & & \leq & 1 \\
 & & & & x_{bc} & & & \leq & 1 \\
 & & & & & x_{bd} & & \leq & 1 \\
 & & & & & & x_{cd} & \leq & 1 \\
 & x_{ab} & +x_{ac} & & +x_{bc} & & & \leq & 2 \\
 & x_{ab} & & +x_{ad} & & +x_{bd} & & \leq & 2 \\
 & & x_{ac} & +x_{ad} & & & +x_{cd} & \leq & 2 \\
 & & & & x_{bc} & +x_{bd} & +x_{cd} & \leq & 2 \\
 & x_{ab}, & x_{ac}, & x_{ad}, & x_{bc}, & x_{bd}, & x_{cd} & \geq & 0
\end{array}
$$



Minimale Spannbäume als Optimierungsproblem

$$
\begin{array}{lrrrrrrl}
\min z = & 3 \cdot x_{ab} & +4 \cdot x_{ac} & +1 \cdot x_{ad} & +5 \cdot x_{bc} & +8 \cdot x_{bd} & +2 \cdot x_{cd} & \\
\text{s.t.} & x_{ab} & +x_{ac} & +x_{ad} & +x_{bc} & +x_{bd} & +x_{cd} & = 3 \\
 & x_{ab} & & & & & & \leq 1 \\
 & & x_{ac} & & & & & \leq 1 \\
 & & & x_{ad} & & & & \leq 1 \\
 & & & & x_{bc} & & & \leq 1 \\
 & & & & & x_{bd} & & \leq 1 \\
 & & & & & & x_{cd} & \leq 1 \\
 & x_{ab} & +x_{ac} & & +x_{bc} & & & \leq 2 \\
 & x_{ab} & & +x_{ad} & & +x_{bd} & & \leq 2 \\
 & & x_{ac} & +x_{ad} & & & +x_{cd} & \leq 2 \\
 & & & & x_{bc} & +x_{bd} & +x_{cd} & \leq 2 \\
 & x_{ab}, & x_{ac}, & x_{ad}, & x_{bc}, & x_{bd}, & x_{cd} & \geq 0
\end{array}
$$

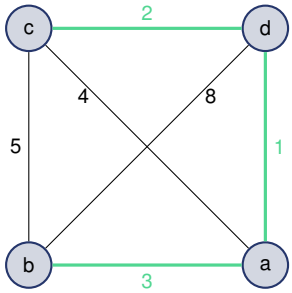


$$
L^* = \begin{pmatrix} x_{ab} \\ x_{ac} \\ x_{ad} \\ x_{bc} \\ x_{bd} \\ x_{cd} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}
$$

$$
z^* = 6
$$

Minimale Spannbäume als Optimierungsproblem

$$
\begin{array}{lrrrrrrl}
\min z = & 3 \cdot x_{ab} & +4 \cdot x_{ac} & +1 \cdot x_{ad} & +5 \cdot x_{bc} & +8 \cdot x_{bd} & +2 \cdot x_{cd} & \\
\text{s.t.} & x_{ab} & +x_{ac} & +x_{ad} & +x_{bc} & +x_{bd} & +x_{cd} & = 3 \\
 & x_{ab} & & & & & & \leq 1 \\
 & & x_{ac} & & & & & \leq 1 \\
 & & & x_{ad} & & & & \leq 1 \\
 & & & & x_{bc} & & & \leq 1 \\
 & & & & & x_{bd} & & \leq 1 \\
 & & & & & & x_{cd} & \leq 1 \\
 & x_{ab} & +x_{ac} & & +x_{bc} & & & \leq 2 \\
 & x_{ab} & & +x_{ad} & & +x_{bd} & & \leq 2 \\
 & & x_{ac} & +x_{ad} & & & +x_{cd} & \leq 2 \\
 & & & & x_{bc} & +x_{bd} & +x_{cd} & \leq 2 \\
 & x_{ab}, & x_{ac}, & x_{ad}, & x_{bc}, & x_{bd}, & x_{cd} & \geq 0
\end{array}
$$



$$
L^* = \begin{pmatrix} x_{ab} \\ x_{ac} \\ x_{ad} \\ x_{bc} \\ x_{bd} \\ x_{cd} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}
$$

$$
z^* = 6
$$