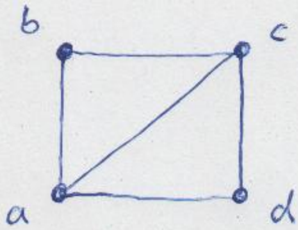
9. Graphen

9.1. Grundbegriffe

Graphen dienen dazu, Netzwerke und Beziehungen zwischen Objekten darzustellen.

Ein Graph $G = (V, E)$ besteht aus einer endlichen nichtleeren Menge V von Knoten und einer Menge E von Kanten, die jeweils zwei Knoten verbinden.

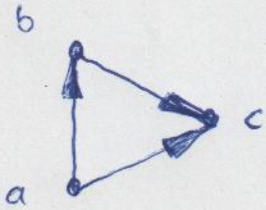
Bsp:



Graph $G = (V, E)$ mit $V = \{a, b, c, d\}$ und $E = \{\{a,b\}, \{b,c\}, \{c,d\}, \{a,d\}, \{a,c\}\}$

Die Kanten eines Graphen können eine Richtung haben: gerichtete Graphen

Bsp:



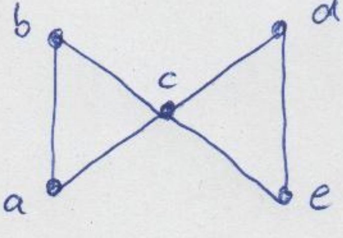
gerichteter Graph $G = (V, E)$ mit $V = \{a, b, c\}$ und $E = \{(a,b), (b,c), (a,c)\}$

Grad eines Knotens v: Anzahl der Kanten, die v als Endpunkt haben

Bezeichnung: $d_G(v)$

Bsp:

G:

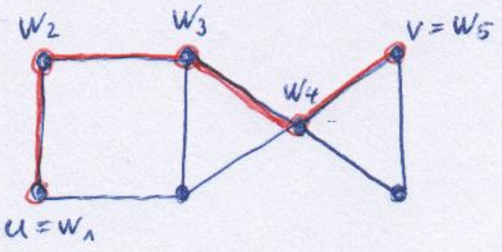


$d_G(c) = 4$

Weg von Knoten u zu Knoten v in G: Folge von paarweise verschiedenen Knoten $w_1, w_2, \ldots, w_k$ mit $u = w_1$ und $v = w_k$, sodass für alle $i \in \{1, 2, \ldots, k-1\}$ eine Kante zwischen w_i und w_{i+1} in G existiert.

Bsp:

G:

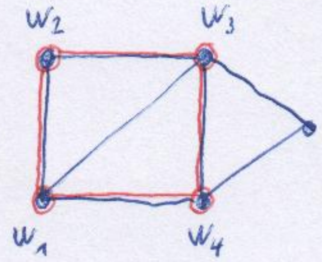


Ein Weg von u nach v in G

Ein Weg $w_1, w_2, \dots, w_k$ mit $k \geq 3$, sodass w_1 und w_k durch eine Kante verbunden sind, heißt Kreis.

Bsp:

G:

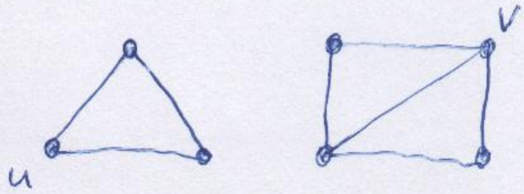


Ein Kreis der Länge 4 in G.

Ein Graph $G = (V, E)$ heißt zusammenhängend, wenn es zwischen allen Knoten $u, v \in V$ einen Weg in G gibt.

Bsp:

G:

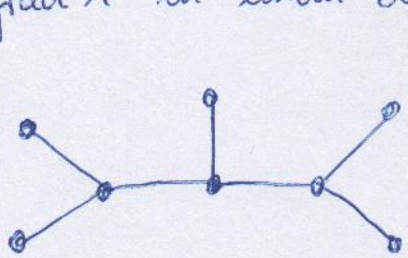


G ist nicht zusammenhängend, da es keinen Weg von u nach v gibt.

9.2. Bäume

Ein zusammenhängender Graph ohne Kreise heißt Baum. Die Knoten vom Grad 1 in einem Baum heißen Blätter.

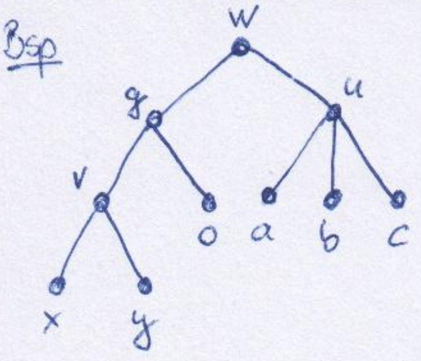
Bsp:



Ein Baum mit 5 Blättern.

Gibt man in einem Baum $T = (V, E)$ einen Knoten $w \in V$ als Wurzel vor, so heißt T gewurzelter Baum. Dadurch erhält man auf den Knoten von T Verwandschaftsgrade:

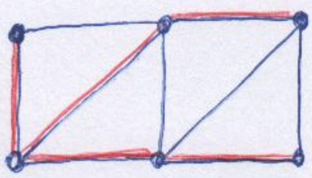
Bsp



- a, b, c sind die Kinder von u
- v ist der Vater von x und y
- g ist der Großvater von x und y
- o ist der Onkel von x und y

Ein Baum, der in einem Graphen $G = (V, E)$ enthalten ist und alle Knoten von G umfasst, heißt Spannbaum von G.

Bsp:



/ Kanten eines Spannbaums

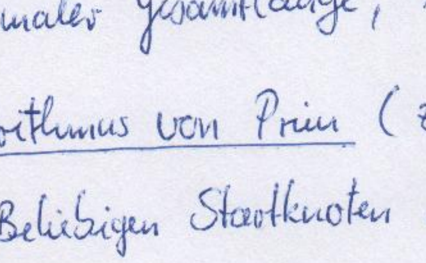
Sind die Kanten $e \in E$ eines Graphen $G = (V, E)$ mit Kantenlängen $l(e)$ versehen, dann sucht man nach einem Spannbaum mit minimaler Gesamtlänge, einem minimalen Spannbaum (MST).

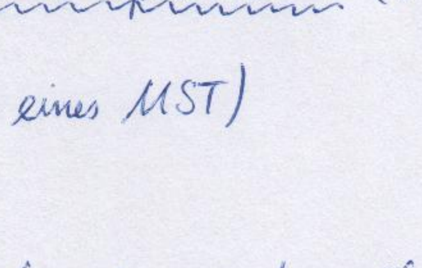
Algorithmus von Prim (zum Berechnen eines MST)

(a) Beliebigen Startknoten s wählen.

(b) Solange noch nicht alle Knoten verbunden sind, wählt man eine Kante mit minimaler Länge, die einen weiteren Knoten anbindet.

Bsp:

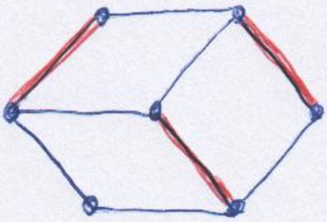
G: 

MST: 

9.3. Matchings

Eine Menge M von Kanten, die paarweise keine gemeinsamen Endpunkte haben, heißt Matching.

Bsp:



/ Kanten eines Matchings

In Anwendungen benötigt man oft ein möglichst großes Matching.

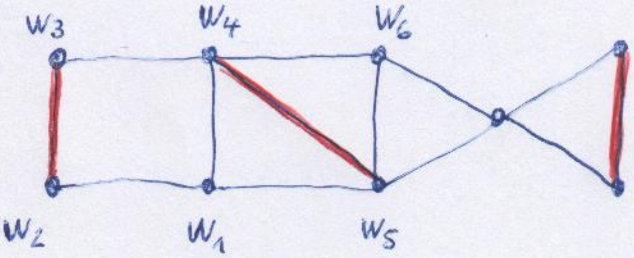
Ein solches findet man schrittweise über augmentierende Wege:

(i) Weg $w_1, w_2, \dots, w_k$

(ii) w_1 und w_k nicht Endpunkte einer Kante im Matching

(iii) $\{w_2, w_3\}, \{w_4, w_5\}, \{w_6, w_7\}, \dots, \{w_{k-2}, w_{k-1}\}$ sind Kanten im Matching

Bsp:

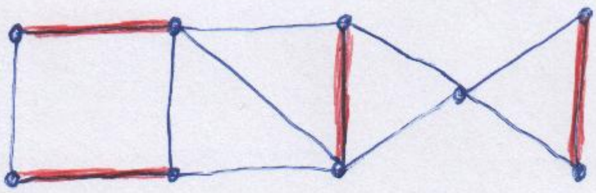


/ Kanten des aktuellen Matchings M

$w_1, w_2, \dots, w_6$ ist augmentierender Weg

Vertauschen der Rollen der Kanten entlang des augmentierenden Weges liefert ein Matching M' mit $|M'| = |M| + 1$.

Bsp: (von oben)



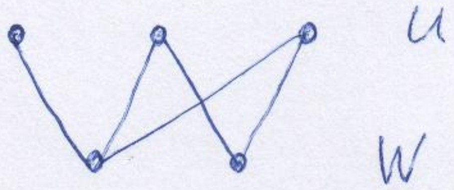
<u>Allgemein gilt:</u>

Es gibt genau dann ein größeres Matching, wenn es einen augmentierenden Weg gibt.

9.4. Bipartite Graphen

Ein Graph $G = (V, E)$ heißt *bipartit*, wenn V in Teilmengen U und W zerlegt werden kann, sodass jede Kante von G genau einen Endpunkt in U und einen in W hat.

Bsp.:

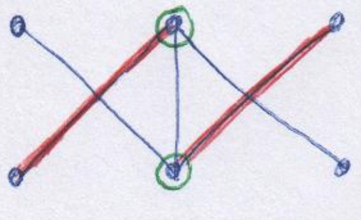


Merke:

(i) Bipartite Graphen sind genau diejenigen, die keine Kreise ungerader Länge haben.

(ii) Jeder Baum ist ein bipartiter Graph.

(iii) In einem bipartiten Graphen ist die Anzahl der Kanten in einem größten Matching immer gleich der Anzahl von Knoten, die gerade noch ausreicht, um von jeder Kante mindestens einen Endpunkt abzudecken (*Vertex-Cover*)



größtes Matching

o kleinstes Vertex-Cover