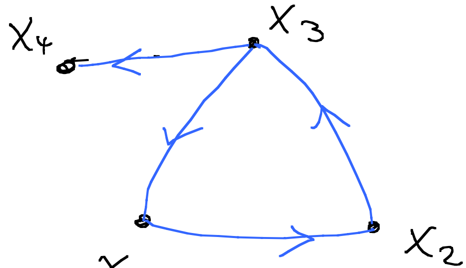
# Vorlesung 27.6.16

Information: Mi 29.6.16 fällt die Vorlesung aus!

## Bedeutung der Adjazenzmatrix

Mit $A$ und Potenzen von $A$ $(A^r)$ sind Aussagen möglich über Anzahl und Existenz von Wegen der Länge $r$ im Digraphen

Bp.:




$$A = \begin{array}{c|cccc} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & 0 & 1 & 0 & 0 \\ x_2 & 0 & 0 & 1 & 0 \\ x_3 & 1 & 0 & 0 & 1 \\ x_4 & 0 & 0 & 0 & 0 \end{array} \qquad A^2 = \begin{array}{c|cccc} & x_1 & x_2 & x_3 & x_4 \\ \hline x_1 & 0 & 0 & 1 & 0 \\ x_2 & 1 & 0 & 0 & 1 \\ x_3 & 0 & \boxed{1} & 0 & 0 \\ x_4 & 0 & 0 & 0 & 0 \end{array}$$

Bp.: Es ex. 1 Weg der Länge 2 von $x_3$ nach $x_2$

$$A^3 = A \cdot A \cdot A = \begin{pmatrix} \boxed{1} & 0 & 0 & 1 \\ 0 & \boxed{1} & 0 & 0 \\ 0 & 0 & \boxed{1} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

Bp.: Es gibt Zyklen der Länge 3 bei $x_1, x_2, x_3$

$$A^4 = \begin{pmatrix} 0 & \boxed{1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

entsprechende Interpretation!

**Def:** Wegematrix $W = (w_{ij})$

$$w_{ij} = \begin{cases} 1 & \text{wenn ein Weg von } x_i \text{ nach } x_j \text{ ex.} \\ 0 & \text{sonst} \end{cases}$$

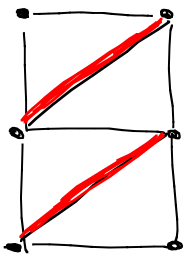
Für das Bsp: $A + A^2 + A^3 + A^4 = \begin{pmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 2 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

$$W = \begin{array}{c|cccc} & x_1 & x_2 & x_2 & x_3 \\ \hline x_1 & & & & \\ x_2 & & \multicolumn{2}{c}{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}} & \\ x_3 & & & & \\ x_4 & & & & \end{array}$$

Wegematrix

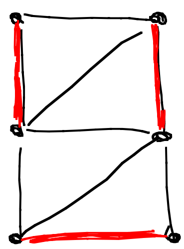
**Einschub:** Teilgebiet der Graphentheorie
Matching

**Def:** Ein Matching in einem Graphen ist eine Menge unabhängiger Kanten, die paarweise nicht adjazent sind.



bipartite Graphen
(2 disjunkte Knotenmengen)

nichterweiterbares
maximales Matching



perfektes Matching
$2|K| = |V|$

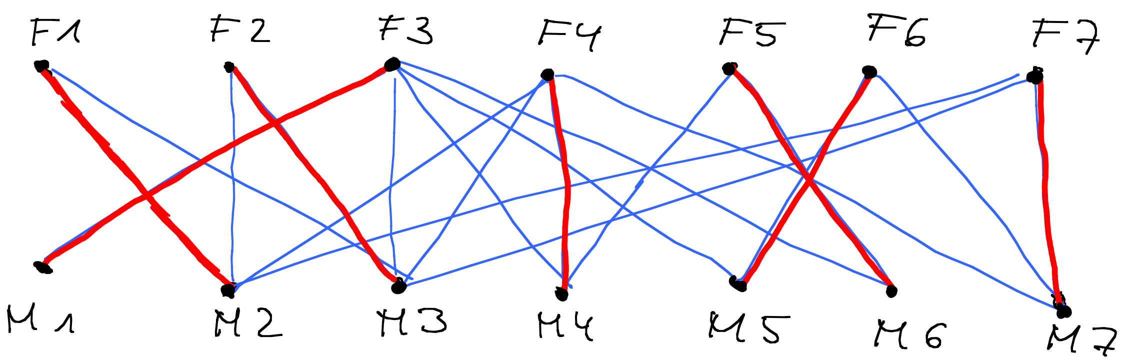
Anwendung: Klassisch: Partnersuche
Jobvermittlung

Bp: Heiratsvermittlung

7 Frauen
7 Männer } jeder erstellt eine Liste seiner Favoriten

Bp:

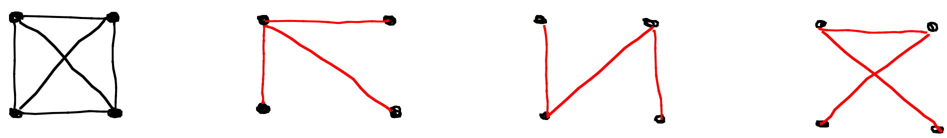
| | Wünsche |
|---|---|
| F1 | M2, M3 |
| F2 | M2, M3 |
| F3 | M1, M3, M4, M5, M6 |
| F4 | M2, M3, M4, M7 |
| F5 | M4, M6 |
| F6 | M5 M7 |
| F7 | M2, M3, M7 |




# Durchlaufen von Graphen

Vor. Kenntnis: Gerüst

Wdh:



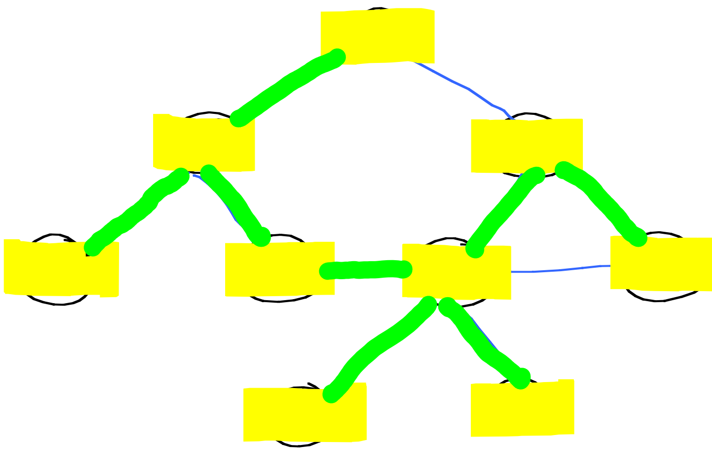
Frage: Wieviele verschiedene Gerüste gibt es hier?
(4 Knoten, 6 Kanten)

$$\binom{6}{3} = \frac{6!}{(6-3)!\cdot 3!} = \frac{6!}{3!\cdot 3!} \doteq \frac{\overset{2}{\cancel{4}}\cdot 5\cdot \overset{2}{\cancel{6}}}{1\cdot \cancel{2}\cdot \cancel{3}} = 20 \text{ Mögl.}$$

d.h. 20 verschiedene Gerüste

Gesucht: Suche mit System

Bp.:



Aufgabe: Suche einen Weg, der jeden Knoten (mindestens) einmal besucht.

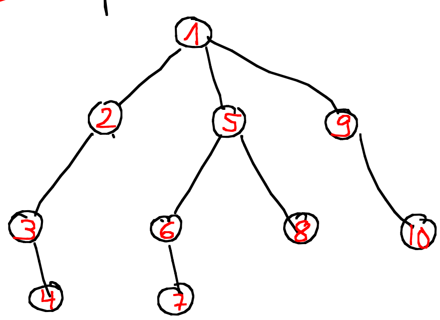
Eine Möglichkeit: Tiefensuche

1) Wähle Startknoten
2) Besuche einen möglichen Nachbarn (bei mehreren einen auswählen (Bp. alphabetisch oder numerisch) Knoten markieren
3) wie 2), falls kein Weg weiterführt, den zuletzt markierten Knoten besuchen! weiter bis alle markiert

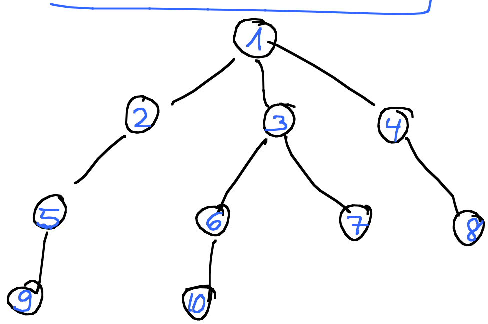
Alternativ: Breitensuche

1) Startknoten festlegen
2) Alle Nachfolger markieren etc.

Bp.: Tiefensuche



Breitensuche



Bei bewerteten Graphen:

Suche nach minimal aufspannenden Bäumen

Def: MST

Ein aufspannender Baum mit minimaler Summe seiner Kantenbewertungen.

Auffinden von MST's:

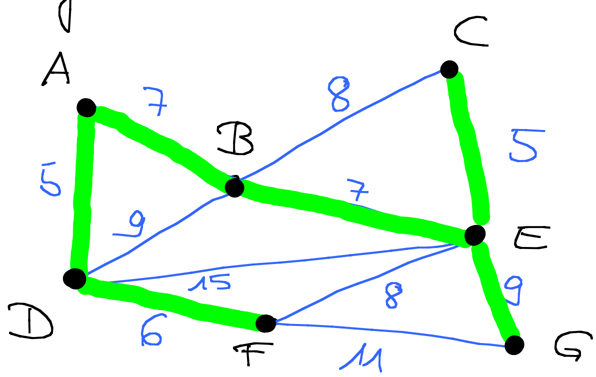
Algorithmus von Kruskal (Joseph Kruskal 1956)

zum Auffinden in ungerichteten bewerteten Graphen

1) Suche die Kante mit dem kleinsten Gewicht, falls Kreis, dann verwerfen sonst hinzufügen

2) 1) so oft wiederholen, bis nichts mehr geht.

Bp:

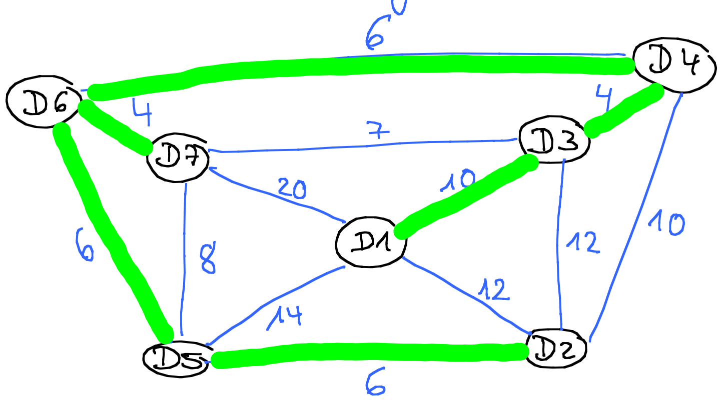



Bp: Sieben Dörfer sollen durch Straßen verbunden werden

Kosten für den Straßenbau:

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|---|---|---|---|---|---|---|---|
| D1 | 0 | 12 | 10 | 0 | 14 | 0 | 20 |
| D2 | 12 | 0 | 12 | 10 | 6 | 0 | 0 |
| D3 | 10 | 12 | 0 | 4 | 0 | 0 | 7 |
| D4 | 0 | 10 | 4 | 0 | 0 | 6 | 0 |
| D5 | 14 | 6 | 0 | 0 | 0 | 6 | 8 |
| D6 | 0 | 0 | 0 | 6 | 6 | 0 | 4 |
| D7 | 20 | 0 | 7 | 0 | 8 | 4 | 0 |

bewertete Adjazenzmatrix




Vorschau auf Vorlesung 4.7

Algorithmus von Dijkstra