



Kapitel 3.3: Zusammenhangskomponenten

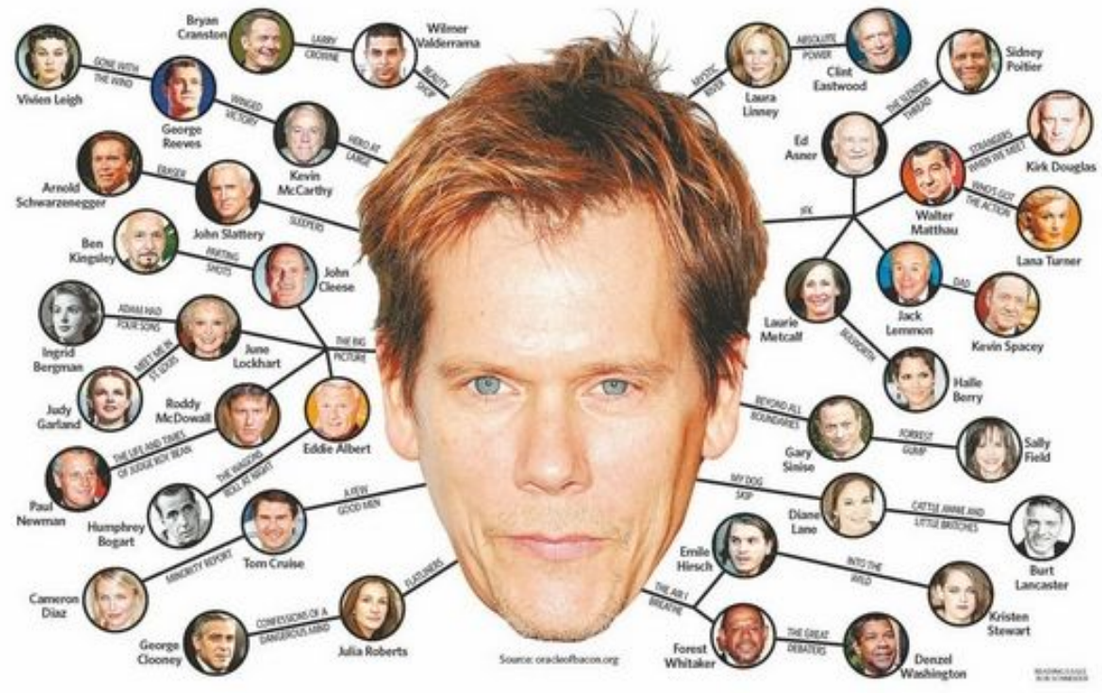
Algorithmen und Datenstrukturen
WS 2026/27

Prof. Dr. Sándor Fekete

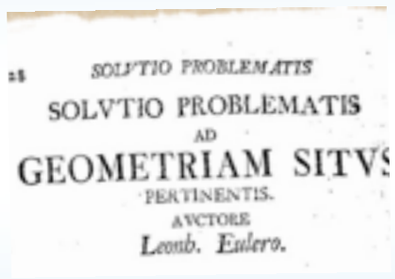
Soziale Vernetzung!

Bitte per Email an fotos-aud@ibr.cs.tu-bs.de:

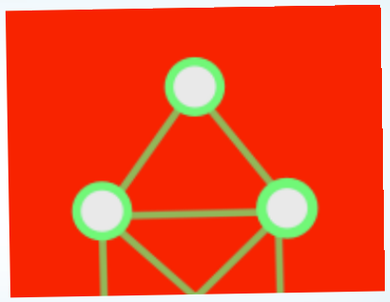
- Erkennbares Bild
- Portrait-Format 4x3
- Dateiname:
  Nachname-Vorname.jpg
  (Bacon-Kevin_Norwood.jpg)
- jpg, png, gif,…
- Optional: „Ich sitze meistens vorne links“ o.ä.






SOLVTIO PROBLEMATIS

SOLVTIO PROBLEMATIS
AD
GEOMETRIAM SITVS
PERTINENTIS.
AVCTORE
Leonh. Eulero.



Kapitel 2: Graphen

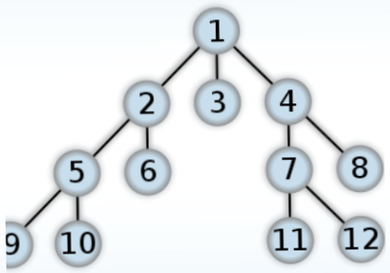
Algorithmen und Datenstrukturen
WS 2026/27

Prof. Dr. Sándor Fekete

Kapitel 2 - Quiz 1

Graphendefinitionen und Eulerwege





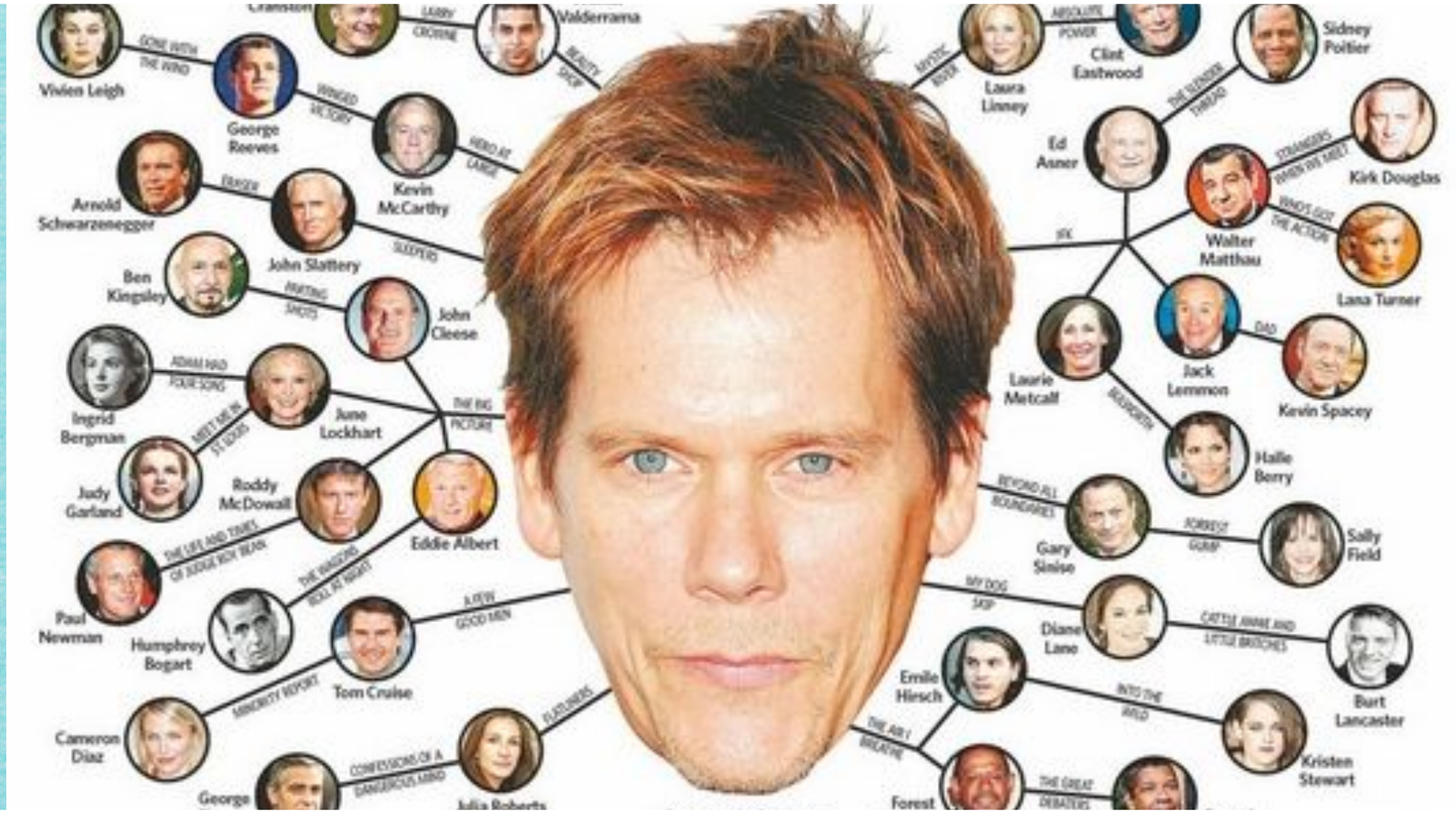


Kapitel 3.3: Zusammenhangskomponenten

Algorithmen und Datenstrukturen
WS 2026/27

Prof. Dr. Sándor Fekete

Problem 3.1: *(s-t-Weg).*
Gegeben: Ein Graph G=(V,E), ein Startknoten s, ein Zielknoten t
Gesucht: Ein Weg von s nach t, falls einer existiert

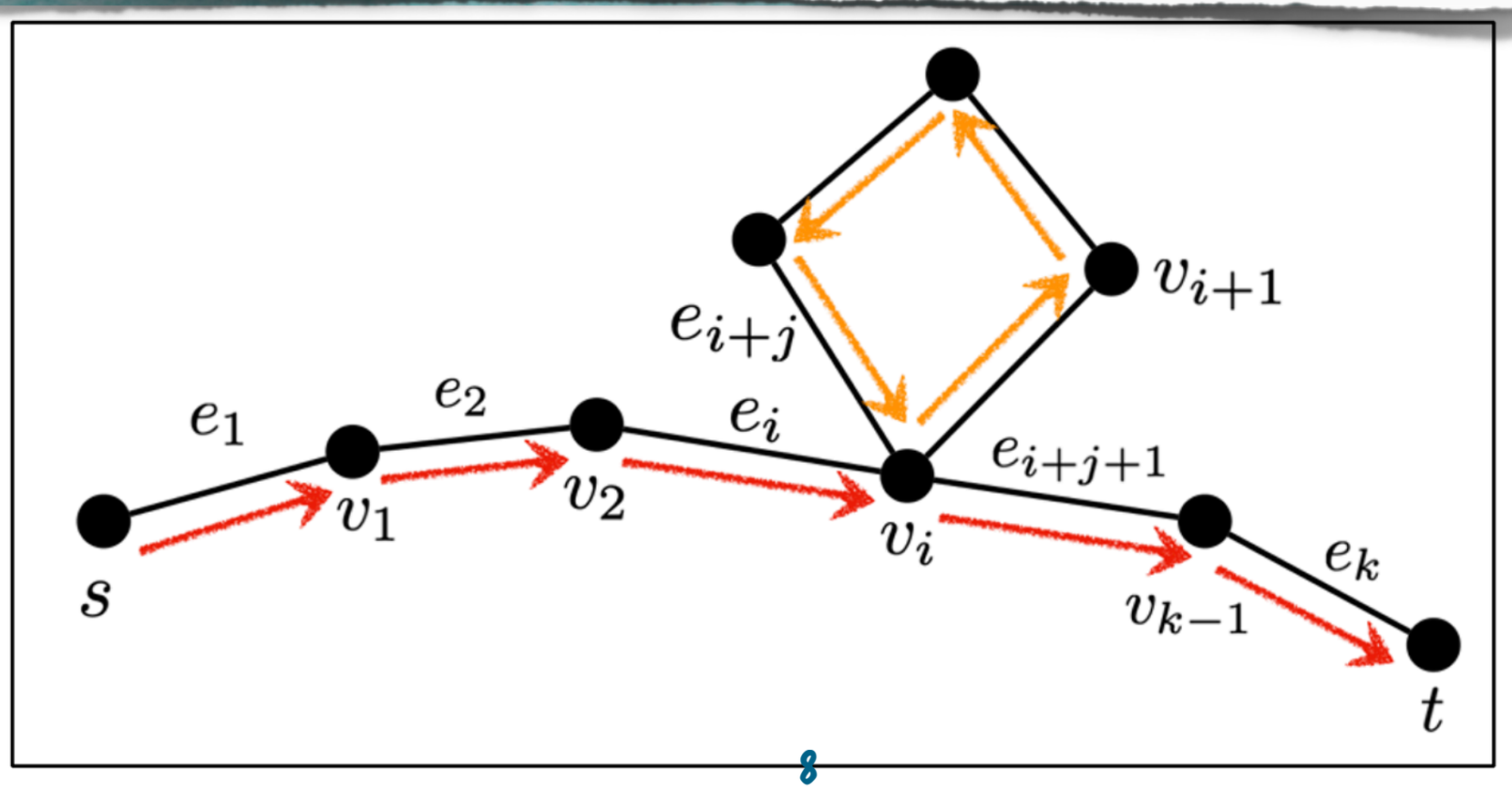


Problem 3.2: *(Zusammenhangskomponente)*
Gegeben: Ein Graph G=(V,E), ein Startknoten s
Gesucht: Die Menge alles von s in G erreichbaren Knoten;
eine Menge von Kanten, die die Erreichbarkeit sicherstellt

Satz 3.3

Wenn ein Weg zwischen zwei Knoten s und t in einem Graphen existiert, dann existiert auch ein Pfad.

$$W^* = s, e_1, v_1, \dots, \underset{\text{erster Besuch}}{v_i}, e_{i+1}, \dots, \underset{\text{letzter Besuch}}{v_i}, e_k, \dots, t$$



Satz 3.3

Wenn ein Weg zwischen zwei Knoten s und t in einem Graphen existiert, dann existiert auch ein Pfad.

$$w^* = s, e_1, v_1, \dots, v_i, e_{i+1}, \dots, v_i, e_k, \dots, t$$

v_i: erster Besuch; v_i: letzter Besuch

Dann gäbe es einen noch kürzeren Weg:

$$w^{**} = s, e_1, v_1, \dots, v_i, e_k, \dots, t$$

v_i: nur einmal besucht!

Das widerspricht der Annahme, dass w^* kürzestmöglich ist.
Also hat w^* keinen doppelt besuchten Knoten, ist also ein Pfad! □

Definition 3.6 (Wald, Baum)

(1) Ein **Wald** ist ein Kreisfreier Graph.

(2) Ein **Baum** ist eine Zusammenhangskomponente in einem Wald.
(Also: ein Kreisfreier, zusammenhängender Graph)

(3) Ein **aufspannender Baum** ist ein Baum, der alle Knoten verbindet. (Manchmal auch: **Spannbaum**. Englisch: "spanning tree")

Algorithmus 3.7

INPUT: Graph G = (V,E), Knoten s

OUTPUT: Knotenmenge $Y \subseteq V$, die von s aus erreichbar ist,

Kantenmenge $T \subseteq E$, die die Erreichbarkeit sicherstellt

1. Sei R:={s}, Y:={s}, T:=∅
2. WHILE (R≠∅) DO {
   - 2.1. Wähle $v \in R$
   - 2.2. IF (es gibt kein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$) THEN
     - 2.2.1. $R := R \setminus \{v\}$
   - 2.3. ELSE {
     - 2.3.1. Wähle ein $w \in V \setminus Y$ mit $e=\{v,w\} \in E$
     - 2.3.2. Setze R := R u {w}, Y := Y u {w}, T := T u {e}
     - }

   }
3. STOP

Satz 3.8. *Der Algorithmus 3.7 ist*

(1) endlich

(2) korrekt.

Mehr demnächst!

s.fekete@tu-bs.de