Universität Augsburg
Institut für Informatik
Lehrstuhl für Organic Computing
Prof. Dr. Jörg Hähner

Ansprechpartner
Dominik Rauh, M. Sc.
dominik.rauh@informatik.uni-augsburg.de
Eichleitnerstr. 30, Raum 502

Wintersemester 2018/2019

Peer-to-Peer und Cloud Computing

Lösungsvorschläge zu Aufgabenblatt 2

1 Verständnisfragen zu Graphentheorie (9 Punkte)

1. Warum sind in P2P-Netzwerken hohe Clustering-Koeffizienten oft erstrebenswert? Geben Sie mehrere Gründe an und erklären Sie diese! (3 Punkte)

   Lösung

   - $C_v \sim$ Anzahl der Verbindungen unter den Nachbarn eines Knotens
   - $C_v = 1 \Leftrightarrow$ Nachbarn von v sind eine *Clique*, also ein total vernetzter Teilgraph
   - $C_v = 0 \Leftrightarrow$ Nachbarn von v haben keine Verbindungen zueinander
   - niedriger $C_v \Rightarrow$ Zufallsgraph um v herum
   - hoher $C_v \Rightarrow$ Small-World–Netzwerk um v herum
   - Gründe:

     höhere Fehlertoleranz redundante Verbindungen $\Rightarrow$ geringere Wahrscheinlichkeit für „Auseinanderbrechen" des Graphen bei Knotenausfall (danach wären Knoten und ihre Ressourcen im jeweils anderen Segment nicht mehr auffindbar!)

     geringere Routing-Komplexität lokal mehr mögliche Pfade $\Rightarrow$ effizienter Zugriff auf nahe Long-Distance–Links etc.

     Performanz durchschnittliche Pfadlänge kürzer $\Rightarrow$ geringere Latenz, schnellere Suche, …

     effizientes Bootstrapping neu hinzugefügte Knoten sind schnell gut eingebunden

2. Weshalb teilt man bei der Berechnung von L_G durch den Term $|V| * (|V| - 1)/2$? Kombinatorik könnte bei der Erklärung helfen! (1 Punkt)

   Lösung

   - L_G ist durchschnittliche Pfadlänge des Graphen G
   - die Summe der Längen aller minimalen Pfade (Distanzen) geteilt durch die Anzahl aller minimalen Pfade
   - Interpretation von $\frac{|V|\ (|V|-1)}{2}$: Anzahl aller minimalen Pfade = Anzahl der Knotenpaare
   - „Händeschüttel-Problem“
     - Wähle einen Knoten v aus $|V|$ Knoten aus.
     - Wähle einen zweiten Knoten aus den verbliebenen $|V - 1|$ Knoten aus (v wird nicht mehr betrachtet, da keine Selbstreferenzen).
     - Eigentlich bidirektionale Kanten wurden jetzt doppelt gezählt $\Rightarrow$ Faktor $\frac{1}{2}$.
   - klassische Kombinatorik: Ziehe zwei Knoten aus $|V|$ ohne Beachtung der Reihenfolge (Bidirektionalität) und ohne Zurücklegen (keine Selbstreferenz)!

   $$\binom{|V|}{2} = \frac{|V|!}{2!\ (|V|-2)!} = \frac{|V|\ (|V|-1)\ (|V|-2)!}{2\ (|V|-2)!} = \frac{|V|\ (|V|-1)}{2}$$

   - Problem: Inseln
     - wenn ein Knoten u nicht von v aus erreichbar ist, dann $d_{v,u} = \infty$
     - somit $L_G = \infty$

   $\Rightarrow$ Aussagekraft von L_G nur bei *verbundenen Graphen* ($\neq$ total/vollständig vernetzt!): Es existiert mindestens ein Pfad zwischen jedem möglichen Paar (v, w) von Knoten

3. In P2P-Netzwerken fehlt es oft an globalem Wissen; jeder Teilnehmer kennt nur die Entfernung zu seinen unmittelbaren Nachbarn. Um L_G zu berechnen, benötigt man aber die Entfernung für jedes Paar von Knoten.

   - Angenommen, man hat bereits für jeden Knoten die Entfernung zu seinen unmittelbaren Nachbarn (dies könnte z. B. die zentrale Einheit in einem zentralisierten P2P-Netzwerk wissen). Welche beiden berühmten Algorithmen eignen sich für die *genaue* Ermittlung der für die Berechnung von L_G benötigten Entfernung für jedes Paar von Knoten? (1 Punkt)

     Lösung

     - Dijkstra und Floyd-Warshall

    - aber auch: Bellman-Ford
    - Breitensuche eher nicht (entspricht Dijkstra mit 1-gewichteten Kanten)
  - Welchen Zusammenhang gibt es zwischen der Anzahl von Kanten im Graphen und der Laufzeit dieser Algorithmen? (3 Punkte)

   Lösung

  - Floyd-Warshall ist besser in dichten (*dense*) Graphen, die fast alle möglichen Kanten haben (denn $O(|V|^3)$)
  - Dijkstra ist besser in dünn besetzten (*sparse*) Graphen, die nur wenige Kanten haben. Dijkstra muss wiederholt durchlaufen (da er immer nur einen Knoten festhält), somit $O(|E|\ |V| + |V|^2 \log |V|)$ (wenn man Fibonacci-Heaps benutzt); bei dünn besetzten Graphen, ist $|E| \ll |V|^2$, somit ist Dijkstra besser.
  - Bellman-Ford: in etwa $O(|E|\ |V|^2)$, vielleicht etwas besser mit Anpassung an „alle Knoten"

4. Ist ein Graph mit $|V| = 1$ und $|E| = 0$ wirklich ein Graph? (1 Punkt)

   Lösung

   Ja, denn die Definition schränkt die Anzahl an Elementen von V und E nicht ein.

2 Rechenaufgaben zu Graphentheorie (12 Punkte)

Beachten Sie bei der Bearbeitung der folgenden Aufgaben diesen Hinweis: Bei den Graphen (c) und (d) gilt, dass – mithilfe von Modulo – mögliche Verbindungen über den „letzten" Knoten v_8 (Graph (c)) bzw. v_{31} (Graph (d)) hinaus betrachtet werden sollen. Beispielsweise wird das Ziel der Kante $\{v_n, v_{n+2^4}\}$ für $n = 25$ wie folgt berechnet: $v_{(25+2^4) \mod 32} = v_9$.

1. Berechnen Sie folgende Werte für die Knoten v_3 und v_4 des Graphen (a), die Knoten v_2 und v_8 des Graphen (b), den Knoten v_1 des Graphen (c) sowie den Knoten v_1 des Graphen (d) (6 Punkte):
   - Grad k_v,
   - Nachbarschaft N_v und
   - Clustering-Koeffizient C_v.

   Lösung

   Graph (a)

   - $k_{v_3} = 3$

- $N_{v_3} = \{2, 4, 5\}$
- $C_{v_3} = \frac{2}{3*(3-1)*\frac{1}{2}} = \frac{2}{3}$
- $k_{v_4} = 4$
- $N_{v_4} = \{1, 3, 5, 6\}$
- $C_{v_4} = \frac{1}{4*(4-1)*\frac{1}{2}} = \frac{1}{6}$

Lösung

Graph (b)

- $k_{v_2} = 2$
- $N_{v_2} = \{1, 3\}$
- $C_{v_2} = \frac{0}{...} = 0$
- $k_{v_8} = 4$
- $N_{v_8} = \{3, 7, 9, 10\}$
- $C_{v_8} = \frac{1}{4*(4-1)*\frac{1}{2}} = \frac{1}{6}$

Lösung

Graph (c)

- $k_{v_1} = 2$
- $N_{v_1} = \{2, 8\}$
- $C_{v_1} = 0$

Lösung

Graph (d)

- $k_{v_1} = 9$
- $N_{v_1} = \{0, 2, 3, 5, 9, 17, 25, 29, 31\}$
- $|N_{v_1}| = 9$
- $C_{v_1} = \frac{12}{(9*(9-1)*\frac{1}{2})} = \frac{1}{3}$

Siehe auch Abbildung 1.

	2	3	5	9	17	25	29	31	0
$2^0 = 1$	3	4	6	10	18	26	30	0	1
$2^1 = 2$	4	5	7	11	19	27	31	1	2
$2^2 = 4$	6	7	9	13	21	29	1	3	4
$2^3 = 8$	10	11	13	17	25	1	5	7	8
$2^4 = 16$	18	19	21	25	1	9	13	15	16

Abbildung 1: Nachbarschaftstabelle für Graph (d). Anzahl der blauen Felder entspricht Anzahl der Verbindungen zwischen Nachbarn von v_1. Die Knoten v_{25}, v_{29}, v_{31} und v_0 besitzen ihrerseits Verbindungen zu v_1 (gelb); aufgrund der Bidirektionalität des Graphen müssen diese Knoten ebenfalls als Nachbarn von v_1 betrachtet werden. Die Verbindung von v_9 zu v_{25} darf nur einmal gezählt werden (Bidirektionalität, rot markiert).

2. Welchen Durchmesser haben die jeweiligen Graphen? (4 Punkte)

   Lösung

   Durchmesser: Der längste aller kürzesten Wege.

   - $D_{(a)} = |p(2,7)| = 4$
   - $D_{(b)} = \infty$ (wegen v_{16})
   - $D_{(c)} = 4$
   - $D_{(d)} = 3$ (siehe Abbildung 2)

3. Berechnen Sie für die Graphen (a) und (b) den Wert L_G. (2 Punkte)

   Lösung

   - $L_{(a)} = \frac{39}{7*(7-1)*\frac{1}{2}} = 1,86$ (siehe Abbildung 3)
   - $L_{(b)} = \infty$

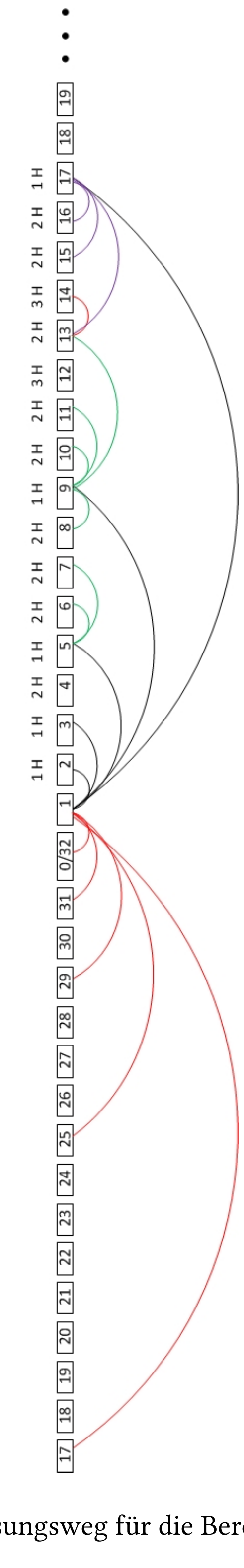
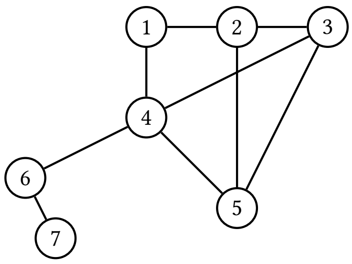


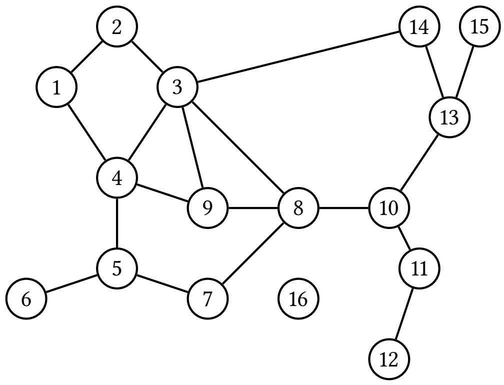
Abbildung 2: Lösungsweg für die Berechnung von $D_{(d)}$.

$v_1 \to v_2$	1										
$v_1 \to v_3$	2	$v_2 \to v_3$	1								
$v_1 \to v_4$	1	$v_2 \to v_4$	2	$v_3 \to v_4$	1						
$v_1 \to v_5$	2	$v_2 \to v_5$	1	$v_3 \to v_5$	1	$v_4 \to v_5$	1				
$v_1 \to v_6$	2	$v_2 \to v_6$	3	$v_3 \to v_6$	2	$v_4 \to v_6$	1	$v_5 \to v_6$	2		
$v_1 \to v_7$	3	$v_2 \to v_7$	4	$v_3 \to v_7$	3	$v_4 \to v_7$	2	$v_5 \to v_7$	3	$v_6 \to v_7$	1

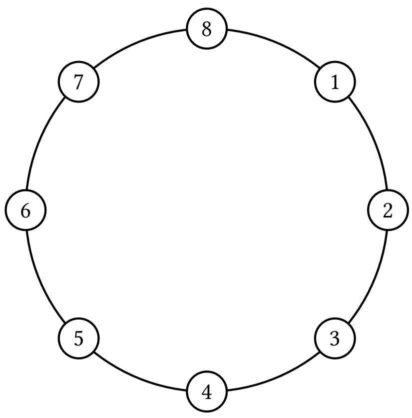
Abbildung 3: Distanzen aller Knotenpaare in Graph (a).



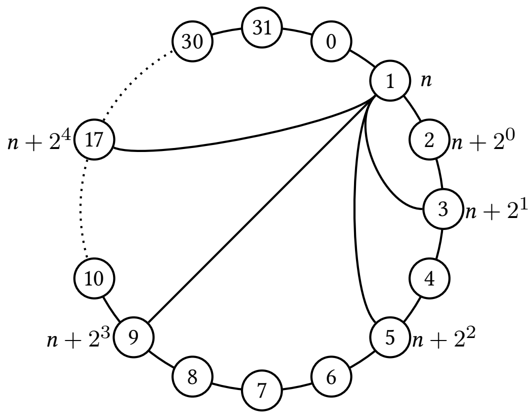
(a)



(b)



(c) Ring



(d) Ring mit Long Distance Connections. Dieser Graph besteht aus 32 Knoten (0 bis 31). Jeder Knoten v_n hat neben der Ringverbindung eine zusätzliche direkte *bidirektionale* Verbindung zu den Knoten v_{n+2^0}, v_{n+2^1}, v_{n+2^2}, v_{n+2^3}, v_{n+2^4}. Achtung: „Doppelte" Kanten werden in Mengen als eine einzige Kante gezählt!

Abbildung 4: Graphen für Aufgabe 2