

# 46114 Herbst 2008

Algorithmen / Datenstrukturen / Programmiermethoden (nicht vertieft)

Aufgabenstellungen mit Lösungsvorschlägen



**Die Bschlangaul-Sammlung**  
Hermine Bschlangaul and Friends

# Aufgabenübersicht

Thema Nr. 1 . . . . .	3
Aufgabe 2 [Dijkstra] . . . . .	3
Thema Nr. 2 . . . . .	5
Aufgabe 3 [Quicksort] . . . . .	5



## Die Bschlangaul-Sammlung

Hermine Bschlangaul and Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

# Thema Nr. 1

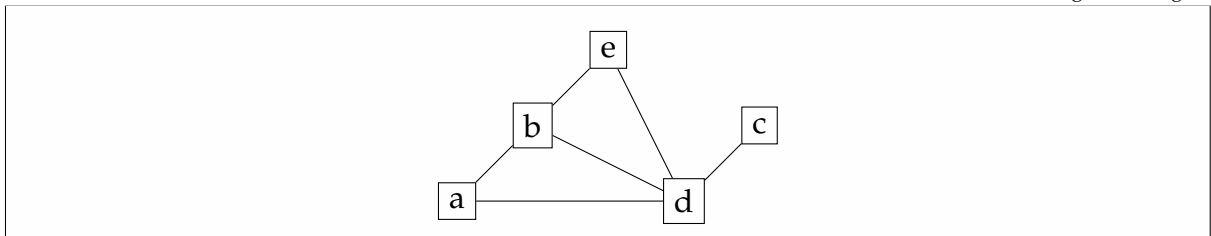
## Aufgabe 2 [Dijkstra]

Gegeben sei folgender Graph:

V: {a, b, c, d, e}  
 E: a → a, b  
 b → b, d, e  
 c → c, d  
 d → a, e

(a) Stellen Sie den Graphen grafisch dar!

Lösungsvorschlag



(b) Berechnen Sie mit dem Algorithmus von Dijkstra schrittweise die Länge der kürzesten Pfade ab dem Knoten a! Nehmen Sie dazu an, dass alle Kantengewichte 1 sind. Erstellen Sie eine Tabelle gemäß folgendem Muster:

ausgewählt | a | b | c | d | e

Ergebnis:

Hinweis: Nur mit Angabe der jeweiligen Zwischenschritte gibt es Punkte. Es reicht also nicht, nur das Endergebnis hinzuschreiben.

Lösungsvorschlag

Nr.	ausgewählt	a	b	c	d	e
1	a	0	1	∞	1	∞
2	b		1	∞	1	2
3	d			2	1	2
4	c			2		2
5	e					2

(c) Welchen Aufwand hat der Algorithmus von Dijkstra bei Graphen mit  $|V|$  Knoten und  $|E|$  Kanten,

- wenn die Kantengewichte alle 1 sind? Mit welcher Datenstruktur und welchem Vorgehen lässt sich der Aufwand in diesem Fall reduzieren (mit kurzer Begründung)?
- wenn die Kantengewichte beliebig sind und als Datenstruktur eine Halde verwendet wird (mit kurzer Begründung)?

# Thema Nr. 2

## Aufgabe 3 [Quicksort]

Quicksort ist ein Sortierungsverfahren, das nach dem Divide-and-Conquer-Prinzip (Teile und Herrsche) arbeitet. Wir betrachten im Folgenden die Anwendung dieses Verfahrens zum Sortieren von Integerzahlen. Die Sortierung soll in aufsteigender Reihenfolge der Werte erfolgen. Wir nehmen dabei an, dass die zu sortierenden Zahlen in einem Feld fester Länge abgelegt sind.

- (a) Beschreiben Sie die Arbeitsweise des Divide-and-Conquer-Prinzips im allgemeinen Fall. Geben Sie dabei die Bedeutung der Schritte divide, conquer und combine an.
- (b) Beschreiben Sie die Arbeitsweise des Algorithmus Quicksort. Geben Sie dabei an, worin die Schritte divide, conquer und combine im konkreten Fall bestehen.
- (c) Geben Sie in C, C++ oder Java eine Implementierung des Algorithmus Quicksort an. Formulieren Sie die Implementierung als rekursive Funktion quicksort() und verwenden Sie das jeweils erste Element. des (Teil-)Feldes für die Aufteilung. Verwenden Sie für Ihre Implementierung von quicksort() «lei Parameter:
  - (i) das Feld, in dem die zu sortierenden Zahlen abgelegt sind;
  - (ii) den Index des am weitesten links gelegenen Elementes des zu sortierenden Teilfeldes;
  - (iii) den Index des am weitesten rechts gelegenen Elementes des zu sortierenden Teilfeldes.

Erläutern Sie die Arbeitsweise Ihrer Implementierung. Kennzeichnen Sie die Schritte divide, conquer und combine des zugrundeliegenden Divide-and-Conquer-Prinzips.