

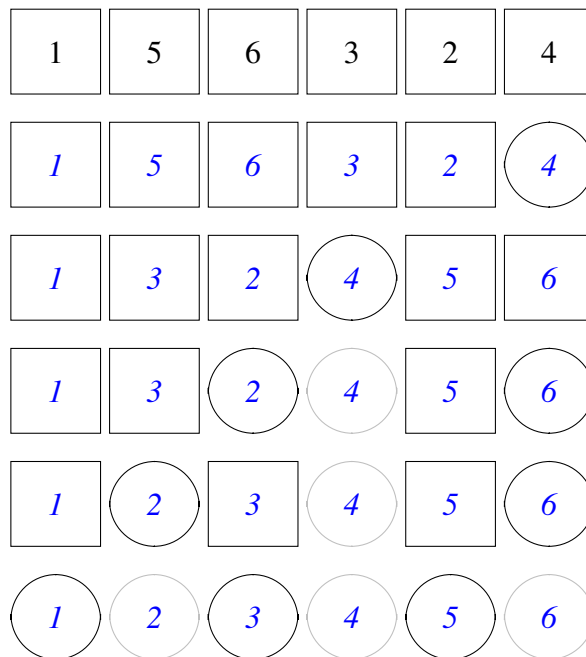
Lösungsvorschläge zur Musterklausur zu Algorithmen und Datenstrukturen

Aufgabe 1 Sortieren mit Quick Sort

(7 Punkte)

Gegeben ist das folgende unsortierte Array, das mit *Quick Sort* sortiert werden soll. Als Pivot-Element soll dabei stets das jeweils letzte Element gewählt werden.

- a) Führen Sie den Quick Sort Algorithmus aus und füllen Sie für jeden Schritt die nachfolgende Verlaufsgrafik aus. Geben Sie dabei jeweils in einer neuen Zeile die Zustände des Arrays nach der Pivotwahl sowie nach dem Partitionierungs-Schritt an! Markieren Sie dazu den Pivot sowie die Grenzen der Teilarrays. Gegebenenfalls überflüssige Zeilen streichen Sie bitte!



In dieser Lösung haben wir die Array-Elemente bei der Partitionierung einfach in ihrer Originalreihenfolge belassen, dies kann man jedoch auch anders machen!

Bewertung: 5 Punkte für den korrekten Ablauf insgesamt. Je 1 Punkt für zur Musterlösung korrespondierende Zeilen. Folgefehler kosten keine Punkte.

- b) Die Laufzeit von Quick Sort hängt stark von den Eingabedaten sowie der gewählten Pivot- und Partitionierungs-Strategie ab. Ordnen Sie Ihr Beispiel zwischen *Best-* und *Worst-Case* ein, begründen Sie Ihre Entscheidung kurz!

Die Antwort hängt massiv von der inneren Sortierung der Partitionen ab und muss dementsprechend zur ersten Teilaufgabe passen. In der Musterlösung bewegt man sich sehr nahe am Best-Case, da bei der gegebenen Pivot-Wahl- und der verwendeten Partitionierungs-Strategie das Ablaufschema stark dem balancierten Binärbaum ähnelt.

Bewertung: 2 Punkte. 1 Punkt für korrekte Einschätzung nahe Best-Case, 1 Punkt für ordentliche Begründung.

Aufgabe 2 KMP-Präfixtabelle

(7 Punkte)

Beim Knuth-Morris-Pratt Algorithmus zur Suche in Zeichenketten spielen Präfixtabellen eine entscheidende Rolle. Vervollständigen Sie die jeweiligen Präfixtabellen zu folgenden Suchmustern!

a)

A	B	C	A	B	D		A	F
-1	0	0	0	1	2	0	0	1

b)

A	B	C	A	B	A	A	B	A	B	C	A
-1	0	0	0	1	2	1	1	2	1	2	3

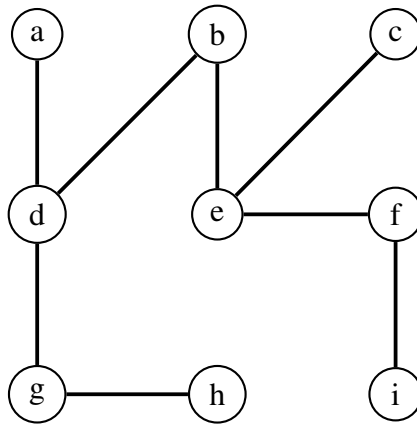
Bewertung: 3 Punkte für korrekte Lösung von a), 1 Punkt Abzug bei einem falschen Eintrag, 0 Punkte nur wenn alles falsch.

4 Punkte für korrekte Lösung von b), 1 Punkt Abzug bei ein oder zwei falschen Einträgen, 0 Punkte nur wenn alles falsch.

Aufgabe 3 Graphen

(8 Punkte)

Gegeben sei der folgende Graph:



a) Füllen Sie die folgende Adjazenzmatrix aus! Lassen Sie dabei Felder, die nicht existierenden Kanten entsprechen, **frei!**

	a	b	c	d	e	f	g	h	i
a				<i>1</i>					
b				<i>1</i>	<i>1</i>				
c					<i>1</i>				
d	<i>1</i>	<i>1</i>					<i>1</i>		
e		<i>1</i>	<i>1</i>			<i>1</i>			
f					<i>1</i>				<i>1</i>
g				<i>1</i>				<i>1</i>	
h							<i>1</i>		
i						<i>1</i>			

- b) Lässt sich dieser Graph einer bestimmten Unterklasse von Graphen zuordnen, und wenn ja, welcher? Geben Sie ausserdem mindestens drei Eigenschaften dieses Graphen an!

Der Graph ist ein nicht-vollständiger Binärbaum mit Wurzel b! Als Eigenschaften sind dementsprechend denkbar: Ungerichtet, ungewichtet, nicht vollständig, zyklensfrei, zusammenhängend, Baumhöhe 3, Binärbaum, ...

Bewertung: 4 Punkte für a), jeweils 0.5 Punkte pro korrekter Kante. 4 Punkte für b), 1 Punkt für Binärbaum, jeweils 1 Punkt für eine korrekte Eigenschaft.

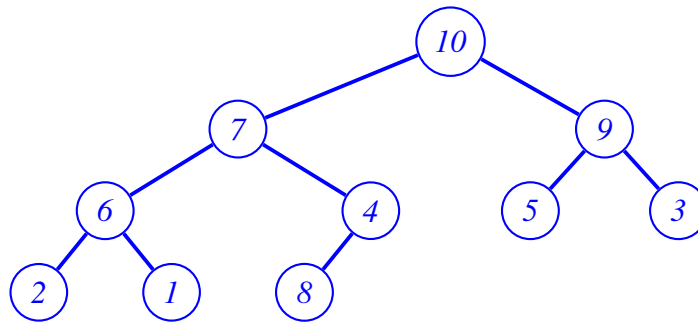
Aufgabe 4 Heaps

(7 Punkte)

- a) Repräsentiert das folgende Array einen Max-Heap? Begründen Sie Ihre Antwort und zeichnen Sie die Baumstruktur! Durch welche konkreten Operationen wäre ggf. die Heap-Eigenschaft herzustellen?

Array: 10, 7, 9, 6, 4, 5, 3, 2, 1, 8

Nein, es ist fast ein Max-Heap, lediglich $8 > 4$ verletzt am letzten Blatt die Ordnung!



*Zur Korrektur **Max-Heapify** ab Knoten 8: Tausche 8 und 4, dann Knoten 8 (neu) und 7. Zu großer Knoten 8 „steigt auf“ bis Kind von 10.*

- b) Kann mittels binärer Suche der **dritt-größte** Wert in einem Max-Heap gefunden werden? Begründen Sie Ihre Antwort!

Nein, ein Heap ist kein binärer Suchbaum, wie er für binäre Suche erforderlich wäre.

Im Suchbaum ist die geforderte Eigenschaft, dass bei jedem Knoten v der größte Wert im linken Teilbaum kleiner/gleich (bzw. der kleinste Wert im rechten Teilbaum größer/gleich) dem Wert bei v selbst ist.

Beim Max-Heap dagegen muss an jedem Knoten v der Wert größer als die Werte seiner beiden unmittelbaren Kinder sein (die Relation gegenüber den ganzen Teilbäumen ergibt sich über die Transitivität!).

Beide Prinzipien sind nicht kompatibel!

Bewertung: 6 Punkte für a), 4 Punkte für korrekten Baum (jeweils 1 Punkt pro Ebene), 1 Punkt für Heap-Eigenschaft verletzt, 1 Punkt für korrektes Aufsteigen von 8.

2 Punkte für b), 1 Punkt für korrekte Antwort, 1 Punkt für ordentliche Begründung.

Aufgabe 5 Dijkstra

(7 Punkte)

Gegeben sei ein gewichteter Graph $G = (V, E)$ mit $V = \{a, b, c, d, e, f\}$, die Kanten und ihre Gewichte sind in der folgende Adjazenzmatrix encodiert. Auf diesem Graphen wird mittels des Dijkstra-Algorithmus ein kürzester Pfad ausgehend von Knoten a gesucht. Der Algorithmus hat bereits einige Schritte durchgeführt. Der aktuelle Zustand der Priority Queue Q ist unten angegeben.

Setzen Sie den Ablauf des Dijkstra-Algorithmus um **zwei** weitere Schritte fort. Geben Sie hierzu die jeweiligen Zustände der Priority Queue Q an!

	a	b	c	d	e	f
a	∞	1	5	∞	∞	∞
b	1	∞	∞	1	1,5	∞
c	5	∞	∞	∞	1	∞
d	∞	1	∞	∞	1	2
e	∞	1,5	1	1	∞	2
f	∞	∞	∞	2	2	∞

Betrachten wir zunächst, wie die Ausgangssituation der Aufgabenstellung erzeugt wurde. Wir starten beim Knoten a und initialisieren die Warteschlange dementsprechend:

$$Q = \{(a/0), (b/\infty), (c/\infty), (d/\infty), (e/\infty), (f/\infty)\}$$

In Schritt 1 wird der Knoten a wegen seines minimalen Schlüsselwertes entfernt, und seine Nachbarn werden (bedingt) aktualisiert durch Summierung des Schlüssels von a (= 0) mit dem jeweiligen Kantengewicht:

$$Q = \{(b/1), (c/5), (d/\infty), (e/\infty), (f/\infty)\}$$

In Schritt 2, dem ersten gefragten Schritt, wird der Knoten b entfernt, und seine Nachbarn werden (bedingt) aktualisiert durch Summierung des Schlüssels von b (= 1) mit dem jeweiligen Kantengewicht. Dabei ergibt sich eine geänderte Reihenfolge:

$$Q = \{(d/2), (e/2,5), (c/5), (f/\infty)\}$$

In Schritt 3, dem zweiten gefragten Schritt, wird der Knoten d entfernt, und seine Nachbarn werden (bedingt) aktualisiert durch Summierung des Schlüssels von d (= 2) mit dem jeweiligen Kantengewicht, wobei diesmal der neu errechnete Wert von e nicht übernommen wird! Dabei ergibt sich wieder eine geänderte Reihenfolge:

$$Q = \{(e/2,5), (f/4), (c/5)\}$$

Bewertung: Jeweils 1 Punkt pro korrektem Eintrag in Q (insgesamt 7 Punkte).

Aufgabe 6 LUP-Zerlegung

(6 Punkte)

Gegeben sei die folgende Matrix $A \in \mathbb{R}^{3 \times 3}$,

$$A = \begin{pmatrix} 0 & 2 & 6 \\ 5 & 4 & 8 \\ 5 & 2 & 0 \end{pmatrix}$$

a) Zeigen Sie, dass die folgenden Matrizen P , L und U die LUP-Zerlegung von A bilden.

$$P = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} 5 & 4 & 8 \\ 0 & 2 & 6 \\ 0 & 0 & -2 \end{pmatrix}$$

$$P \cdot A = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 2 & 6 \\ 5 & 4 & 8 \\ 5 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 5 & 4 & 8 \\ 0 & 2 & 6 \\ 5 & 2 & 0 \end{pmatrix}$$

$$L \cdot U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 5 & 4 & 8 \\ 0 & 2 & 6 \\ 0 & 0 & -2 \end{pmatrix} = \begin{pmatrix} 5 & 4 & 8 \\ 0 & 2 & 6 \\ 5 & 2 & 0 \end{pmatrix}$$

$$\Rightarrow P \cdot A = L \cdot U$$

□

Bewertung: 3 Punkte insgesamt, 1 Punkt für korrektes Produkt PA , 1 Punkt für korrektes Produkt LU , 1 Punkt für korrekte Formel $PA = LU$.

b) Betrachten Sie nun den LUP-Algorithmus. Folgende Abbildung zeigt zuletzt den Zustand nach Pivotsierungsschritt 1. Führen Sie den LUP-Algorithmus bis zum Ende durch, und geben Sie dabei alle weiteren Schritte als markierte Matrix an (wie in der Vorlesung)!

$$\begin{aligned} & \left(\begin{array}{c|ccc} 1 & 0 & 2 & 6 \\ 2 & 5 & 4 & 8 \\ 3 & 5 & 2 & 0 \end{array} \right) \xrightarrow{\text{Pivotsierung}} \left(\begin{array}{c|ccc} 2 & 5 & 4 & 8 \\ 1 & 0 & 2 & 6 \\ 3 & 5 & 2 & 0 \end{array} \right) \xrightarrow{\text{Teilerlegung}} \\ & \xrightarrow{\text{Pivotsierung}} \left(\begin{array}{c|ccc} 2 & 5 & 4 & 8 \\ 1 & 0 \{0/5\} & 2 \{2-4 \cdot 0\} & 6 \{6-8 \cdot 0\} \\ 3 & 1 \{5/5\} & -2 \{2-4 \cdot 1\} & -8 \{0-8 \cdot 1\} \end{array} \right) \xrightarrow{\text{Teilerlegung}} \\ & \xrightarrow{\text{Teilerlegung}} \left(\begin{array}{c|ccc} 2 & 5 & 4 & 8 \\ 1 & 0 & 2 & 6 \\ 3 & 1 & -2 & -8 \end{array} \right) \xrightarrow{\text{Teilerlegung}} \\ & \xrightarrow{\text{Teilerlegung}} \left(\begin{array}{c|ccc} 2 & 5 & 4 & 8 \\ 1 & 0 & 2 & 6 \\ 3 & 1 & -1 \{-2/2\} & -2 \{-8-6 \cdot (-1)\} \end{array} \right) \end{aligned}$$

In geschweiften Klammern $\{\dots\}$ ist dabei stets die Berechnungsvorschrift angegeben zur leichteren Nachvollziehbarkeit. Dies wird in der Fragestellung nicht erwartet!

Bewertung: 3 Punkte insgesamt, jeweils 1 Punkt für Teilerlegung und Pivotsierung.

Aufgabe 7 Least-Squares

(8 Punkte)

Gegeben seien die folgenden Datenpunkte.

$$\{(-2/5), (0/1), (1/2)\}$$

- a) Berechnen Sie unter Angabe der Rechenschritte die quadratische Least-Squares-Ausgleichskurve $F(x)$! Wie lautet der Fachausdruck für die Ausgangs-Matrix?

Berechnung via Vandermonde-Matrix V :

$$\underbrace{\begin{pmatrix} 1 & -2 & 4 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}}_{=: V} \cdot \underbrace{\begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}}_{=: \mathbf{a}} = \underbrace{\begin{pmatrix} 5 \\ 1 \\ 2 \end{pmatrix}}_{=: \mathbf{y}}$$

Lösung via Gauss-Elimination des erweiterten Systems $(V|\mathbf{y})$:

$$\begin{array}{ccc} \begin{pmatrix} 1 & -2 & 4 & | & 5 \\ 1 & 0 & 0 & | & 1 \\ 1 & 1 & 1 & | & 2 \end{pmatrix} & \xrightarrow{I \leftrightarrow II} & \begin{pmatrix} 1 & 0 & 0 & | & 1 \\ 1 & -2 & 4 & | & 5 \\ 1 & 1 & 1 & | & 2 \end{pmatrix} & \xrightarrow{II-I, III-I} \\ \rightsquigarrow \begin{pmatrix} 1 & 0 & 0 & | & 1 \\ 0 & -2 & 4 & | & 4 \\ 0 & 1 & 1 & | & 1 \end{pmatrix} & \xrightarrow{II+2 \cdot III} & \begin{pmatrix} 1 & 0 & 0 & | & 1 \\ 0 & 0 & 6 & | & 6 \\ 0 & 1 & 1 & | & 1 \end{pmatrix} & \xrightarrow{II/6, II \leftrightarrow III} \\ \rightsquigarrow \begin{pmatrix} 1 & 0 & 0 & | & 1 \\ 0 & 1 & 1 & | & 1 \\ 0 & 0 & 1 & | & 1 \end{pmatrix} & \xrightarrow{II-III} & \begin{pmatrix} 1 & 0 & 0 & | & 1 \\ 0 & 1 & 0 & | & 0 \\ 0 & 0 & 1 & | & 1 \end{pmatrix} & & \\ & & \Rightarrow F(x) = x^2 + 1 & & \end{array}$$

Man hätte die Elimination natürlich auch vorher, also bei Dreiecksform, abbrechen können, und ab dort mittels eines Substitutionsverfahrens fortfahren.

- b) Zusätzlich zu den gegebenen Datenpunkten sollen nun weitere (potentiell verrauschte) Datenpunkte hinzukommen. Was für eine Art von Lösung würde man nun berechnen, und geben Sie dafür auch eine Formel an. Wie würde man diese Art von Lösung dann tatsächlich berechnen? Geben Sie auch hierfür eine Formel sowie ihren Namen an.

Bei mehr Punkten wird das System $V\mathbf{a} = \mathbf{y}$ überbestimmt. Daher berechnet man nun eine Least squares Lösung:

$$\min_{\mathbf{a}} \|V\mathbf{a} - \mathbf{y}\|^2$$

Die Lösung berechnet sich dann durch die Normalengleichung:

$$V^T V \mathbf{a} = V^T \mathbf{y}$$

Bewertung: 4 Punkte für a). 1 Punkt für korrektes Aufstellen der Matrix-Gleichung, 1 Punkt für Name Vandermonde-Matrix, 1 Punkt für korrektes Lösen, 1 Punkt für korrekte Schlussfolgerung $F(x) = x^2 + 1$.

4 Punkte für b). 1 Punkt für Least-Squares Lösung, 1 Punkt für korrekt Formel dazu. 1 Punkt für Normalengleichung, 1 Punkt für korrekte Formel.