

Lösungsvorschläge zur Musterklausur zu Algorithmen und Datenstrukturen

Aufgabe 1 Sortieren mit Heap Sort

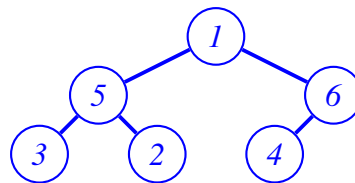
(12 Punkte)

Die sequentielle Liste

1, 5, 6, 3, 2, 4

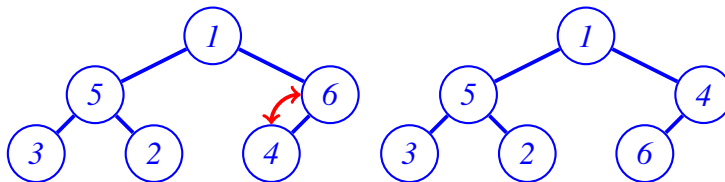
soll mit (*Min*-)Heap Sort sortiert werden!

a) Geben Sie den Zustand der Liste während der **buildMinHeap** nach jedem **minHeapify** an!



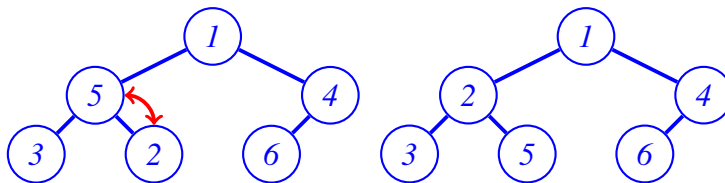
- **minHeapify(4), minHeapify(2), minHeapify(3):**
 ~> Blätter, nichts zu tun
 => Liste nach jedem **minHeapify** jeweils: (1, 5, 6, 3, 2, 4)

- **minHeapify(6):**



=> Liste: (1, 5, 4, 3, 2, 6)

- **minHeapify(5):**



=> Liste: (1, 2, 4, 3, 5, 6)

- **minHeapify(1):**

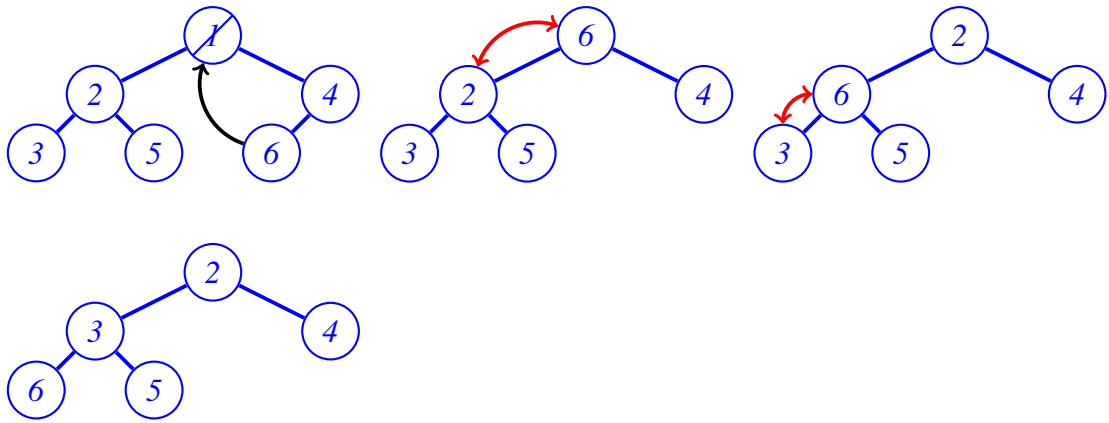
~> nichts zu tun

=> Liste: (1, 2, 4, 3, 5, 6)

Bewertung: 6 Punkte für den korrekten Ablauf insgesamt. Je 1 Punkt für korrekte Angabe der Liste.

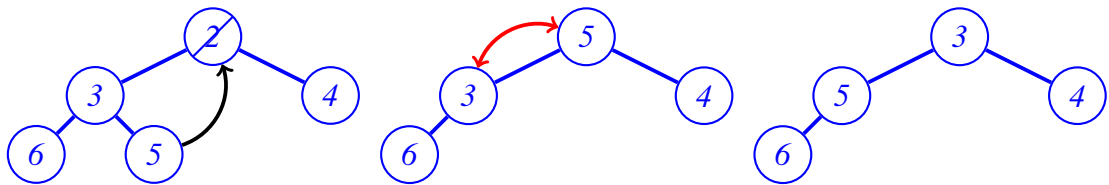
b) Geben Sie den Zustand der Liste während der Sortierung nach jedem **extractMin** an!

- **extractMin = 1, dadurch 6 neue Wurzel und minHeapify(6)**



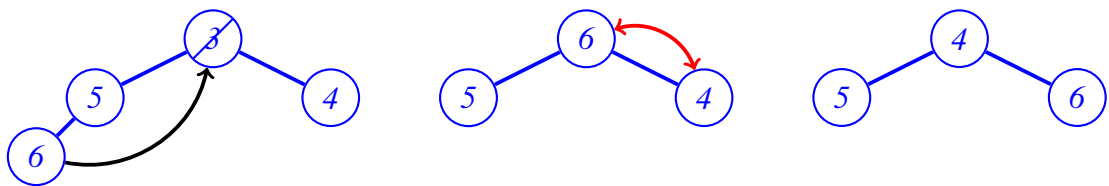
⇒ Liste nach **extractMin**: (2, 3, 4, 6, 5 | 1)

- **extractMin** = 2, dadurch 5 neue Wurzel und **minHeapify**(5)



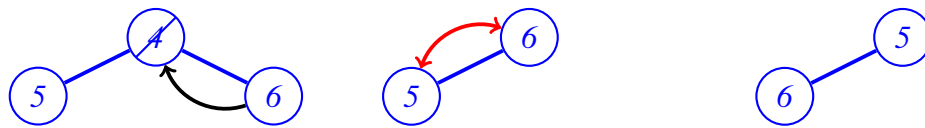
⇒ Liste nach **extractMin**: (3, 5, 4, 6 | 2, 1)

- **extractMin** = 3, dadurch 6 neue Wurzel und **minHeapify**(6)



⇒ Liste nach **extractMin**: (4, 5, 6 | 3, 2, 1)

- **extractMin** = 4, dadurch 6 neue Wurzel und **minHeapify**(6)



⇒ Liste nach **extractMin**: (5, 6 | 4, 3, 2, 1)

- **extractMin** = 5, dadurch 6 neue Wurzel und **minHeapify**(6)



⇒ Liste nach **extractMin**: (6 | 5, 4, 3, 2, 1)

- **extractMin** = 6, dadurch Heap leer



⇒ Liste nach *extractMin*: (6,5,4,3,2,1)

Bewertung: 6 Punkte für den korrekten Ablauf insgesamt. Je 1 Punkt für korrekte Angabe der Liste.

Aufgabe 2 KMP-Präfixtabelle

(7 Punkte)

Geben Sie die jeweiligen Präfixtabellen aus dem Knuth-Morris-Pratt Algorithmus zu folgenden Suchmustern an!

a) "ABCABD AF"

-1,0,0,0,1,2,0,0,1

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

b) "ABCABAABABCA"

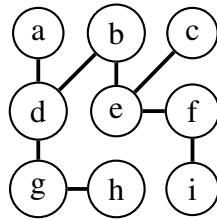
-1,0,0,0,1,2,1,1,2,1,2,3

Bewertung: 4 Punkte für korrekte Lösung, 1 Punkt Abzug bei einem oder zwei falschen Einträgen, 0 Punkte nur wenn alles falsch.

Aufgabe 3 Graphen

(8 Punkte)

Gegeben sei der nebenstehende Graph:



a) Geben Sie die Adjazenzmatrix an!

0	0	0	1	0	0	0	0	0
0	0	0	1	1	0	0	0	0
0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	1	0	0
0	1	1	0	0	1	0	0	0
0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0

Bewertung: 4 Punkte insgesamt, jeweils 0.5 Punkte pro korrekter Kante.

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 insgesamt, 1 Punkt für Binärbaum, jeweils 1 Punkt für eine korrekte Eigenschaft.

Aufgabe 4 Heaps

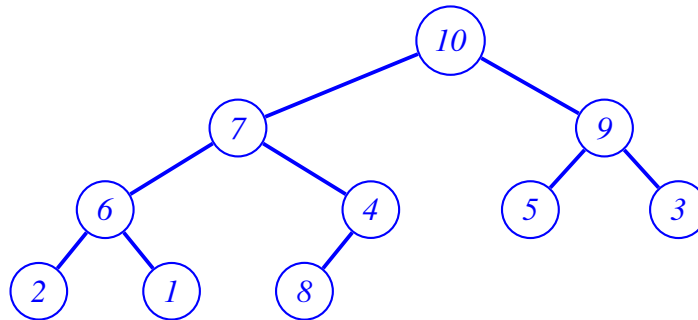
(8 Punkte)

a) Repräsentiert das folgende Array einen Max-Heap?

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

Begründen Sie Ihre Antwort und zeichnen Sie die Baumstruktur! Durch welche konkreten Operationen wäre ggf. die Heap-Eigenschaft herzustellen?

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.*

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

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:** 2 Punkte insgesamt, 1 Punkt für korrekte Antwort, 1 Punkt für ordentliche Begründung*

Aufgabe 5 Dijkstra

(7 Punkte)

Gegeben sei ein gewichteter Graph $G = (V, E)$ mit $V = \{a, b, c, d, e, f\}$ und Adjazenzmatrix:

	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	∞

Mittels des Dijkstra-Algorithmus wird ein kürzester Pfad von Knoten a gesucht, und einige Schritte wurden bereits durchgeführt. Setzen Sie den Ablauf um **zwei** weitere Schritte fort und geben Sie hierzu die jeweiligen Zustände der Priority Queue Q an!

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)!

$$\begin{pmatrix} 1 & 0 & 2 & 6 \\ 2 & 5 & 4 & 8 \\ 3 & 5 & 2 & 0 \end{pmatrix} \xrightarrow{\text{Pivotsierung}} \begin{pmatrix} 2 & 5 & 4 & 8 \\ 1 & 0 & 2 & 6 \\ 3 & 5 & 2 & 0 \end{pmatrix} \xrightarrow{\text{Teilerlegung}}$$

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

$$\xrightarrow{\sim} \begin{pmatrix} 2 & 5 & 4 & 8 \\ 1 & 0 & 2 & 6 \\ 3 & 1 & -2 & -8 \end{pmatrix} \xrightarrow{\text{Teilerlegung}}$$

$$\xrightarrow{\sim} \left(\begin{array}{c|cc|cc} 2 & 5 & 4 & 8 \\ 1 & 0 & 2 & 6 \\ 3 & 1 & -2 & -8 \end{array} \right) \xrightarrow{\sim}$$

$$\xrightarrow{\sim} \left(\begin{array}{c|cc|cc} 2 & 5 & 4 & 8 \\ 1 & 0 & 2 & 6 \\ 3 & 1 & -1 \{-2/2\} & -2 \{-8-6 \cdot (-1)\} \end{array} \right)$$

In geschweiften Klammern {...} 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 $(V|\mathbf{y})$:

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

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

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

- 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 insgesamt, 1 Punkt für Least-Squares Lösung, 1 Punkt für korrekt Formel dazu, 1 Punkt für Normalengleichung, 1 Punkt für korrekte Formel