Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Thomas Seidl
Anna Beer, Florian Richter

SoSe 2018
23.07.2018

Klausur
Algorithmen und Datenstrukturen

Vorname:	
Name:	
Matr.-Nr.:	

Die Klausur besteht aus 7 Aufgaben. Die Punktezahl ist bei jeder Aufgabe angegeben. Bitte überprüfen Sie, ob Sie ein vollständiges Exemplar erhalten haben.

Tragen Sie die Lösungen in den dafür vorgesehenen Raum im Anschluss an jede Aufgabe ein. Falls der Platz für Ihre Lösung nicht ausreicht, benutzen Sie bitte nur die ausgeteilten Zusatzblätter!
Tragen Sie bitte oben auf jeder ungeraden Seite Ihren Namen und Ihre Matrikelnummer ein.

Verwenden Sie keinen Rot-, Grün- oder Bleistift!

Aufgabe	mögliche Punkte	erreichte Punkte
1. Allgemeine Fragen	21	
2. Gemischte Aufgaben	11	
3. Suchbäume	10	
4. Sortieralgorithmen	8	
5. Graphalgorithmen	14	
6. Komplexität	8	
7. Paradigmen	13	
Summe:	85	
Note:		

Entwertung der Klausur

Durch meine Unterschrift bestätige ich, dass meine Klausur **nicht korrigiert** und **nicht gewertet** werden soll. Falls meine Prüfungsordnung keine Entwertungsregel enthält, so wird meine Klausur als **nicht bestanden** gewertet.

München, den 23.07.2018 Unterschrift: _____

Aufgabe 1 Allgemeine Fragen (21 Punkte)

Bitte schreiben Sie Ihre Antworten für die folgenden Fragen *in* die vorgegebenen Kästchen, Rechenwege und Begründungen werden nicht gewertet. Jede (Teil-)Aufgabe gibt einen Punkt, falls nicht anders angegeben.

(a) Was ist die bestmögliche Worst-Case-Laufzeit von Sortieralgorithmen in O-Notation in Abhängigkeit von der Anzahl n der zu sortierenden Objekte?

(b) Welche Eigenschaft wird hier jeweils beschrieben?

i. "Für die gleiche Eingabe wird stets die gleiche Ausgabe berechnet (aber andere Zwischenzustände sind möglich)."

ii. "Der Algorithmus läuft für jede Eingabe nur endlich lange."

iii. "Für die gleiche Eingabe ist die Ausführung und die Ausgabe stets identisch."

(c) Welche Operation gibt das oberste Element eines Stacks aus und entfernt es zugleich?

(d) Wie nennt man einen Baum der Arität 2?

(e) Die Höhe h sei die Anzahl der Kanten auf dem längsten von der Wurzel ausgehenden Pfad.

i. Wie viele Blätter hat ein Baum der Höhe h mit Arität a **maximal**?

ii. Wie viele Knoten hat ein Baum der Höhe h mit Arität 3 mindestens?

(f) Welche Komplexität hat die Matrixaddition von zwei Matrizen der Größe $n \times n$?

(g) Welche Komplexität hat die Matrixmultiplikation von zwei Matrizen der Größen $l \times m$ und $m \times n$? (2P)

(h) Nennen Sie zwei der drei möglichen Reihenfolgen, die durchlaufenen Knoten bei einem Tiefendurchlauf auszugeben. Verwenden Sie die Fachbegriffe. (2P)

(i) Können die folgenden Knoten in der gegebenen Reihenfolge bei der Suche nach der Zahl 77 in einem binären Suchbaum, der Zahlen zwischen 1 und 100 speichert, besucht worden sein? Antworten Sie jeweils entweder mit ja oder mit nein. Falls Ihre Antwort "nein" lautet, *umkreisen Sie den ersten ausschlaggebenden Knoten.* (4P)

i. 10,60,50,80,77 ☐

ii. 33,99,44,66,77 ☐

iii. 12,23,34,45,77 ☐

iv. 1,99,20,80,77 ☐

(j) Nennen Sie drei der vier Eigenschaften, die eine totale Ordnung beschreiben. Verwenden Sie die Fachbegriffe. (3P)

☐ ☐ ☐

(k) Was berechnet der Jarvis' March Algorithmus?

☐

Aufgabe 2 Gemischte Aufgaben (2+4+2+3 Punkte)

(a) Gegeben sei das Universum $U = 0, 1, 2, 3, 4$.

i. Geben Sie den Bitvektor für die Menge $M_1 = 0, 3, 4$ an.

ii. Geben Sie den Bitvektor für die Menge $M_2 = 0, 1, 2, 3$ an.

(b) Gegeben ist die Hashfunktion $h(k) = k \mod 5$. Fügen Sie nacheinander folgende Schlüssel in die untenstehenden Hashtabellen ein: 2, 15, 18, 7, 13

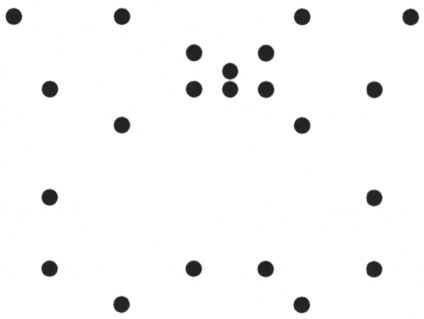
i. Verwenden Sie dabei offenes Hashing.

0	•
1	•
2	•
3	•
4	•

ii. Verwenden Sie dabei geschlossenes Hashing mit linearem Sondieren und $c_1 = 1$.

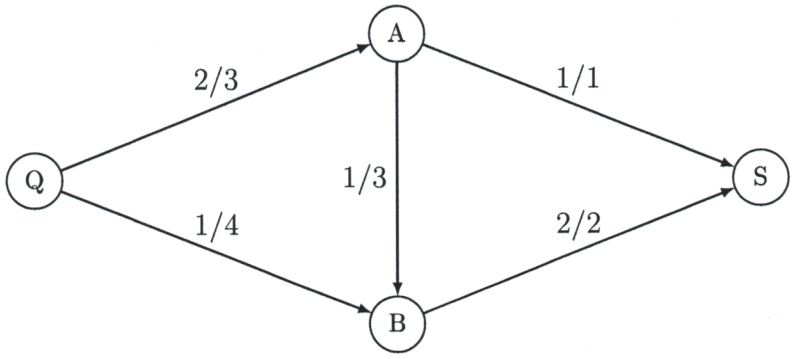
0	•
1	•
2	•
3	•
4	•

(c) Zeichnen Sie die konvexe Hülle zu der gegebenen Punktmenge ein.

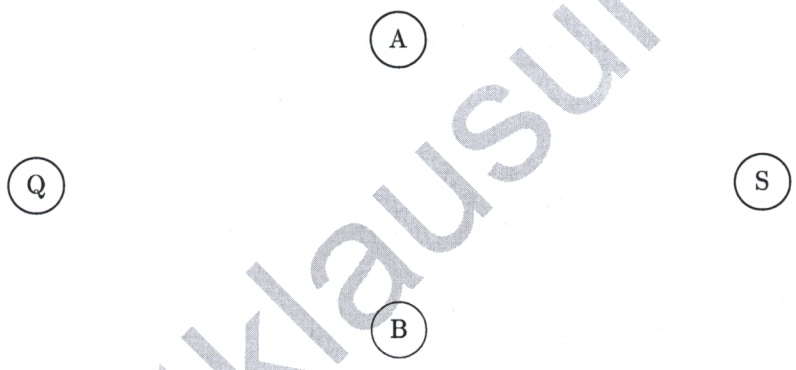


(d) Zeichnen Sie das Residualnetzwerk zu dem gegebenen Flussnetzwerk in die Vorlage ein.

Flussnetzwerk:



Residualnetzwerk:

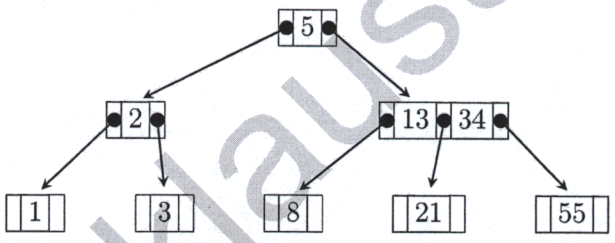


Aufgabe 3 Suchbäume (2+4+4 Punkte)

(a) Geben Sie den binären Suchbaum an, der entsteht, wenn man folgende Elemente der Reihe nach einfügt:

16 13 5 18 19 17 15

(b) Gegeben ist folgender B-Baum der Ordnung $k = 1$:



Geben Sie in den folgenden Teilaufgaben lediglich das Endresultat an und starten Sie jeweils erneut mit dem oben angegebenen B-Baum!

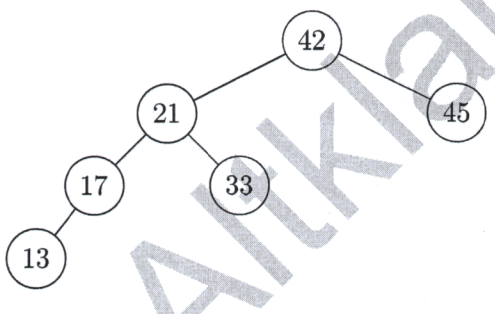
i. Fügen Sie in den oben gegebenen B-Baum den Wert 4 und den Wert 89 ein.

ii. Entfernen Sie aus dem ursprünglich in der Aufgabenstellung angegebenen B-Baum den Wert 3.

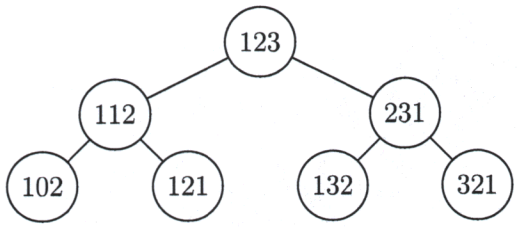
(c) Gegeben sind die folgenden Bäume. Entscheiden Sie jeweils, ob der gegebene Baum ein AVL-Baum ist oder nicht.

Begründen Sie Ihre Aussage, falls es kein gültiger AVL-Baum ist!

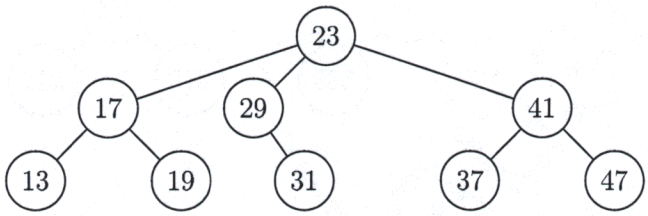
i.



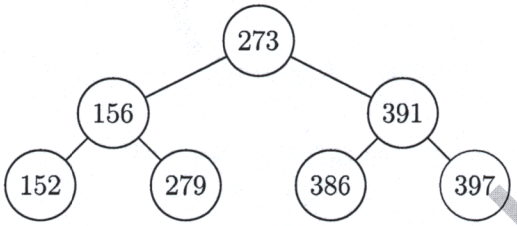
ii.



iii.

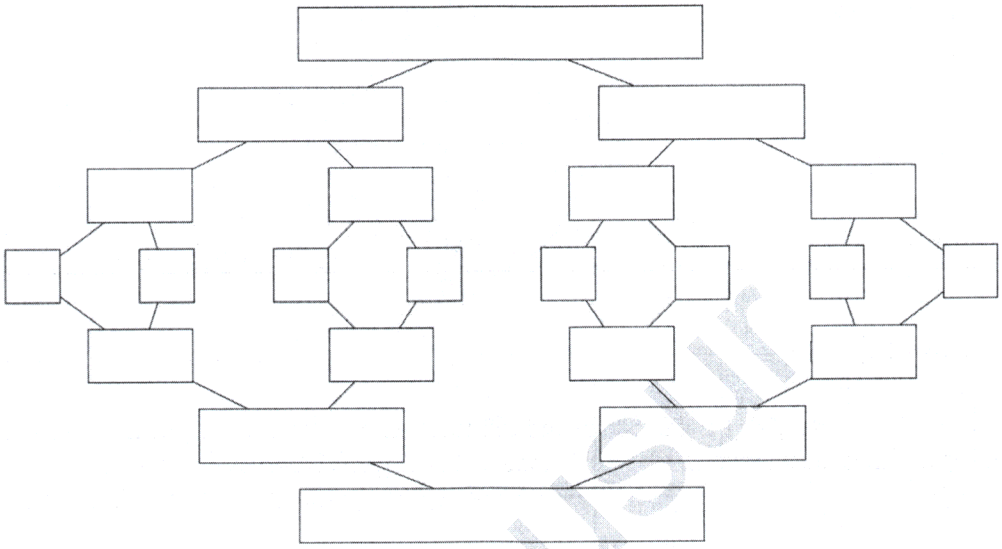


iv.



Aufgabe 4 Sortieralgorithmen (6+1+1 Punkte)

(a) Gegeben sei folgende Liste von Zahlen: 35, 16, 73, 97, 25, 54, 37, 14.
Sortieren Sie die Liste mit **MergeSort**. Geben Sie dabei alle Zwischenschritte an, indem Sie die unten gegebene Vorlage ausfüllen.

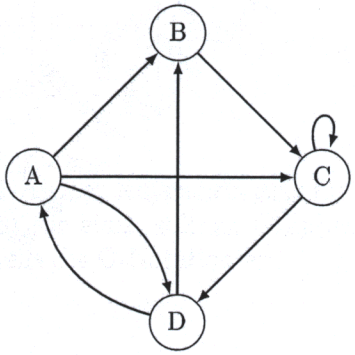


(b) Welches algorithmische Paradigma setzt MergeSort um?

(c) Geben Sie die Worst-Case-Laufzeit von MergeSort an.

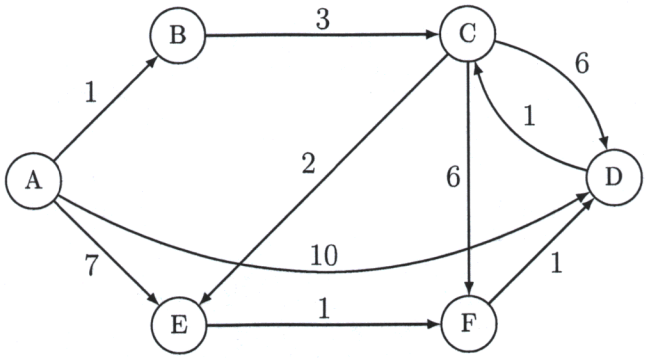
Aufgabe 5 Graphalgorithmen (4+2+5+3 Punkte)

(a) Betrachten Sie den folgenden gerichteten, ungewichteten Graphen. Geben Sie den Graphen als Adjazenzmatrix und als Adjazenzliste an. Verwenden Sie die unten gegebene Tabelle für die Adjazenzmatrix.



(b) Wie hoch ist der Speicherbedarf einer Adjazenzmatrix für einen Graphen mit n Knoten und m Kanten? Wie hoch ist für einen solchen Graphen der Speicherbedarf für eine Adjazenzliste? Geben Sie jeweils die O-Notation an.

(c) Betrachten Sie den folgenden gerichteten Graphen mit Kosten gemäß Kantenbeschriftungen. Haben Sie im Folgenden in einem Schritt eine gleichwertige Wahlmöglichkeit zwischen zwei Knoten, so gehen Sie *alphabetisch* vor.

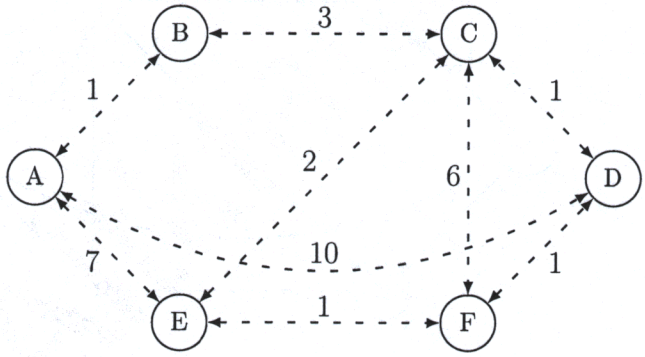


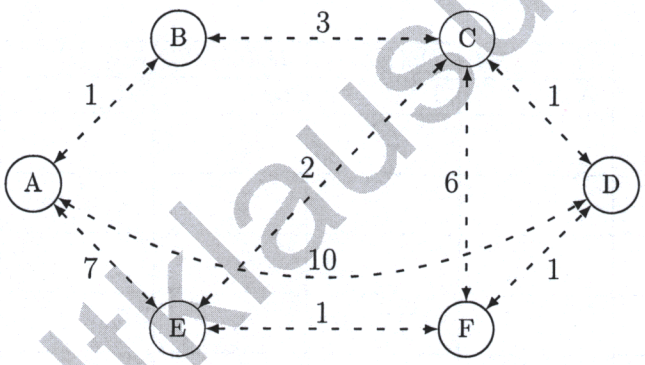
Ermitteln Sie mittels Dijkstra-Algorithmus die Kosten der günstigsten Verbindungen von Knoten A aus. v_k bezeichnet den in Iteration k ausgewählten Knoten, $D_k(X)$ die bisher kürzeste Distanz von Knoten A zu X.

k	v_k	$D_k(B)$	$D_k(C)$	$D_k(D)$	$D_k(E)$	$D_k(F)$
0	A	1	∞	10	7	∞

Hinweis: Es stehen mehr Zeilen zur Verfügung als benötigt werden.

(d) Ermitteln Sie mittels Prim-Algorithmus den minimalen Spannbaum des Graphen. Beginnen Sie mit Knoten A. **Zeichnen** Sie die Kanten ein und **nummerieren** Sie sie anhand ihrer Einfügereihenfolge. Nutzen Sie die Ersatzvorlage, falls Ihr erster Lösungsversuch fehlgeschlagen ist.





Aufgabe 6 Komplexität (5+3 Punkte)

(a) Bestimmen Sie die Komplexität der binären Suche mit dem Mastertheorem. Geben Sie die entsprechenden Werte für alle Variablen der (unten gegebenen) Rekursionsgleichung an.

Hinweis: Für eine Rekursionsgleichung der Form

$$T(n) \leq \begin{cases} c & , n \leq 1 \\ aT(\frac{n}{b}) + n^d & , n > 1 \end{cases}$$

mit $c > 0, a > 0, b > 1, d \geq 0$ *gilt:*

$$T(n) \in \begin{cases} \mathcal{O}(n^d) & , d > \log_b a \\ \mathcal{O}(n^d \log n) & , d = \log_b a \\ \mathcal{O}(n^{\log_b a}) & , d < \log_b a \end{cases}$$

$a =$ [ ] , $b =$ [ ] , $c =$ [ ] , $d =$ [ ]

$\Longrightarrow T(n) \in \mathcal{O}($ [ ] $)$

(b) Zur Lösung des Rucksackproblems lassen sich naiv alle Kombinationen ausprobieren. Die Laufzeit dafür ist $\mathcal{O}(2^n)$. Nach einigen Tests für $n = 5$ wurden aber nie mehr als 30 Vergleiche benötigt. Ist dies ein Widerspruch? Begründen Sie.

Aufgabe 7 Paradigmen (3+6+4 Punkte)

(a) Beschreiben Sie knapp die Strategie des Backtracking. Benennen Sie ein konkretes geeignetes Problem für diese Klasse.

(b) Bestimmen Sie die Edit-Distanz zwischen den Wörtern „Halle“ und „Haus“. Füllen Sie dazu die folgende Tabelle. Markieren Sie deutlich die Edit-Distanz als Ergebnis und zeichnen Sie den Weg entlang dem man zu dieser kommt ein.

(c) Formulieren Sie das Rucksackproblem. Was sind die Eingangsgrößen? Was ist die gesuchte Lösung?

Ergänzung zu Aufgabe: _____

Ergänzung zu Aufgabe: _____

Ergänzung zu Aufgabe: _____