Lehrgebiet
Theoretische
Informatik

Klausur 1
Datenstrukturen und Algorithmen

24.08.2021

VORNAME: NAME:

MATRIKELNUMMER: STUDIENGANG:

Hinweise:

- Als einziges Hilfsmittel ist ein *unkommentiertes* Wörterbuch für Deutsch-Deutsch bzw. Deutsch-Englisch zugelassen. Dieses wird von der Aufsicht während der Klausur kontrolliert.
- Mit der Unterschrift erklären Sie, dass Sie sich gesundheitlich in der Lage fühlen, diese Klausur mitzuschreiben.
- Jedes Lösungsblatt ist mit Name und Matrikelnummer zu versehen.
- Schreiben Sie Ihre Lösungen zu einer Aufgabe, falls vorhanden, in die dafür vorgesehene Box. Falls der Platz nicht ausreicht, kennzeichnen Sie bitte deutlich, wo die Lösung zu finden ist.
- Bitte schreiben Sie deutlich. Unleserliches wird nicht korrigiert und nicht gewertet.
- Schmierblätter werden mit abgegeben; streichen Sie diese durch oder machen Sie sie anderweitig als Schmierblätter kenntlich. Es kann nur ein Lösungsversuch pro Aufgabe gewertet werden. Im Zweifel wird das Falsche gewertet.
- Bitte verwenden Sie einen dokumentenechten Stift mit blauer oder schwarzer Tinte und verwenden Sie keinen Tintenkiller, Tipp-Ex("White-out") oder Ähnliches. Benutzen Sie nur das zur Verfügung gestellte Papier. Entfernen Sie nicht die Heftklammern.
- Sie haben 120 Minuten zur Bearbeitung der Klausur.
- Bewahren Sie Ihre Taschen vor sich (nach Möglichkeit in der Reihe vor Ihnen) auf dem Boden, halten Sie diese während der gesamten Klausur geschlossen. Ein Zugriff (auch auf die geschlossene) Tasche kann als Täuschungsversuch gewertet werden.
- Sie dürfen sich innerhalb der Klausuraufgaben auf Ergebnisse aus der Vorlesung und Ergebnisse aus den Übungsaufgaben beziehen. Dies bedeutet jedoch nicht, dass Sie eine gesamte Aufgabe alleine mit "Wurde bereits in der Vorlesung gezeigt" o.Ä. beantworten dürfen.

Ich versichere, die Klausur selbstständig bearbeitet zu haben. Mir ist bekannt, dass die Klausur bei einem Täuschungsversuch mit „nicht bestanden“ bewertet wird.

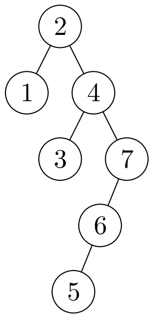
....
(Unterschrift)

Aufgabe	1	2	3	4	Σ
Punkte	30	30	30	30	120
erreicht					

Note: ....

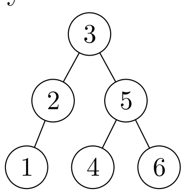
Aufgabe K1 (8 + 6 + 8 + 8 = 30 Punkte)

a) Gegeben sei folgender Splaybaum:



Geben Sie jeweils die resultierenden Splaybäume graphisch an die entstehen, wenn Sie nach dem Verfahren aus der Vorlesung aus diesem Splaybaum zuerst den Schlüssel 2 und aus dem daraus resultierenden Baum den Schlüssel 7 löschen.

b) Wir betrachten nun folgenden Splaybaum:



Geben Sie eine Einfügereihenfolge (ohne Duplikate) der Knoten an, durch die dieser Splaybaum entsteht.

c) Betrachten Sie folgenden Algorithmus.

```
public static int[] sort(int[] input) {
  SplayTree⟨Integer⟩ tree = new SplayTree⟨⟩();
  for(int i = 0; i < input.length; i++) {
    tree.add(input[i]);
  }
  int[] output = new int[input.length];
  for(int i = 0; i < input.length; i++) {
    int min = tree.min();
    tree.remove(min);
    output[i] = min;
  }
  return output;
}
```

Wir benutzen hier einen Splay-Baum, um ein Array zu sortieren. Gehen Sie davon aus, dass die Operation `min` den minimalen Schlüssel ausgibt und amortisiert genau so schnell wie das Suchen in einem Splay-Baum ist. Geben Sie für diesen Algorithmus eine möglichst gute, asymptotische, worst-case Laufzeit an und begründen Sie ihre Antwort. (*Keine Punkte ohne Begründung!*)

d) Geben Sie graphisch einen gerichteten Graphen mit höchstens 5 Knoten an, bei welchem eine Tiefensuche im Tiefensuchbaum eine Querkante erzeugt. Geben Sie den Tiefensuchbaum an und markieren Sie deutlich die Querkante.

Aufgabe K2 (23 + 4 + 3 = 30 Punkte)
Gegeben sind folgende Schlüssel mit dazugehörigen Zugriffswahrscheinlichkeiten: A(0.1), E(0.2), I(0.2), O(0.2), U(0.3). Die Schlüssel sind in alphabetischer Reihenfolge aufsteigend geordnet: $A < E < I < O < U$. Konstruieren Sie einen optimalen Suchbaum wie folgt:

a) Füllen Sie untenstehende Tabellen für $w_{i,j}$ und $e_{i,j}$ nach dem Verfahren aus der Vorlesung aus. Geben Sie in $e_{i,j}$ ebenfalls alle möglichen Wurzeln des optimalen Suchbaums für $\{i, \ldots, j\}$ an. Sie dürfen dazu die Notation aus der Übung verwenden.

$w_{i,j}$	A	E	I	O	U
A					
E	—				
I	—	—			
O	—	—	—		
U	—	—	—	—	

$e_{i,j}$	A	E	I	O	U
A	( )	( )	( )	( )	( )
E	—	( )	( )	( )	( )
I	—	—	( )	( )	( )
O	—	—	—	( )	( )
U	—	—	—	—	( )

b) Geben Sie einen optimalen Suchbaum für die Schlüssel A, E, I, O, U mit den gegebenen Zugriffswahrscheinlichkeiten und der gegebenen Reihenfolge der Schlüssel graphisch an.

c) Ist der optimale Suchbaum für die Schlüssel A, E, I, O, U mit den gegebenen Zugriffswahrscheinlichkeiten und der gegebenen Reihenfolge der Schlüssel eindeutig? Geben Sie dazu eine kurze Begründung an.

Sie finden im Anschluss an die Aufgabe weitere Kopien der obigen Tabellen für ihre Notizen.

Kopien der Tabellen aus Aufgabe K2:

$w_{i,j}$	A	E	I	O	U
A					
E	—				
I	—	—			
O	—	—	—		
U	—	—	—	—	

$e_{i,j}$	A	E	I	O	U
A	( )	( )	( )	( )	( )
E	—	( )	( )	( )	( )
I	—	—	( )	( )	( )
O	—	—	—	( )	( )
U	—	—	—	—	( )

$w_{i,j}$	A	E	I	O	U
A					
E	—				
I	—	—			
O	—	—	—		
U	—	—	—	—	

$e_{i,j}$	A	E	I	O	U
A	( )	( )	( )	( )	( )
E	—	( )	( )	( )	( )
I	—	—	( )	( )	( )
O	—	—	—	( )	( )
U	—	—	—	—	( )

Aufgabe K3 (15 + 15 = 30 Punkte)

a) Sei (Q, S) mit $S \subseteq 2^Q$, wobei $Q \subseteq \mathbf{Q}$ eine endliche Menge rationaler Zahlen ist. Sei weiterhin $T \in S$ genau dann, wenn $\sum_{t \in T} t \leq 1$. Beweisen oder widerlegen Sie: Für alle Q gilt: (Q, S) ist ein Matroid.

Kreuzen Sie bitte an: ☐ Die Aussage gilt. ☐ Die Aussage gilt nicht.

b) Sei $G = (V, E)$ ein ungerichteter Graph. Die unabhängigen Mengen $S \subseteq 2^E$ sind genau die Kantenmengen $T \subseteq E$, deren induzierter Graph maximal einen Kreis enthält. Beweisen oder widerlegen Sie: (E, S) ist ein Matroid.

Kreuzen Sie bitte an: ☐ Die Aussage gilt. ☐ Die Aussage gilt nicht.

Aufgabe K4 (9 + 21 = 30 Punkte)

a) Gegeben sei folgende Hashfunktion: $x \mapsto x \bmod 100$. Fügen sie folgende Schlüssel der Reihe nach in einen anfangs leeren Treap ein und verwenden Sie die Hashfunktion, um die Priorität jedes Schlüssels zu bestimmen, indem Sie jeweils den Schlüssel selbst hashen:

$142, 194, 430, 1900, -24, 13$

Geben Sie den Treap nach jedem Einfügen graphisch an.

b) Geben Sie einen Algorithmus (in Textform oder ausführlich kommentiertem Pseudocode) an, der eine *split(x)*-Funktionalität implementiert und in $O(h)$, wobei h die Höhe des Treaps ist, läuft. *split(x)* sollte den Treap in zwei Treaps aufteilen, einer der nur Elemente größer x und der andere der nur Elemente kleiner gleich x enthält. Begründen Sie die Korrektheit und die Laufzeit ihres Verfahrens.

Tipp: Wenden Sie ihr Verfahren beispielsweise mit dem Aufruf *split(125)* auf den Treap aus Aufgabenteil a) an, um zu testen, ob ihr Algorithmus so funktioniert, wie Sie es erwarten. (Hinweis: dieser Tipp ist zur eigenen Kontrolle sinnvoll, wird allerdings nicht mit Punkten bedacht)

Datenstrukturen und Algorithmen, SS 2021
Prof. Dr. P. Rossmanith
S. Dollase, I. Fesefeldt, M. Gehnen, H. Lotze

Datum: 24.08.2021

Klausur 1

Datenstrukturen und Algorithmen, SS 2021
Prof. Dr. P. Rossmanith
S. Dollase, I. Fesefeldt, M. Gehnen, H. Lotze

Datum: 24.08.2021

Klausur 1