

Probeproofung Angabe Algorithmen und Datenstrukturen

Allgemeine Hinweise:

- Laden Sie sich zusätzlich zu dieser Angabe die Textdatei mit dem Titel "loesungsbogen.txt" als Vorlage für Ihre Abgabe herunter, in die Sie Ihre Lösungen zu den Aufgaben der Prüfung eintragen.
- Nutzen Sie bitte ausschließlich die .txt-Datei ("loesungsbogen.txt"), um ihre Lösung abzugeben.
- Die Art und Weise, eine Aufgabe zu lösen, wird Ihnen detailliert auf der Angabe beschrieben. Lesen Sie diese daher bitte genau.
- Sie bekommen ab Beginn der Prüfung 180 Minuten Zeit, diese herunterzuladen, zu bearbeiten und Ihre Lösung wieder hochzuladen (alles via Uni2work). Somit haben Sie 90 Minuten extra Zeit, um auf eventuelle technische Probleme reagieren zu können.
- Speichern Sie vor Bearbeitung der Klausur die Lösungsdatei mit dem Titel "loesungsbogen_[Matrikelnummer].txt" an einem Ort ab, an dem Sie sie jederzeit wiederfinden. Ersetzen Sie dabei [Matrikelnummer] mit ihrer Matrikelnummer.
- Vergessen Sie nicht, die Lösungsdatei möglichst häufig Zwischenzuspeichern. So reduzieren Sie Stress durch technische Probleme am Ende der Abgabezeit.
- Notfall-Hotline: 089 / 2180 9306

- Ich habe die Lösung eigenständig und ohne die Hilfe Dritter angefertigt.
- Ich bin der rechtmäßige Nutzer dieses Uni2Work-Accounts und gebe die Lösung nicht für jemand Drittes ab.
- Ich bin zum Zeitpunkt der Prüfung immatrikuliert und kann dies auch durch ein entsprechendes Dokument jederzeit während des Prüfungsprozesses bestätigen.
- Ich veröffentliche keinerlei Prüfungsdokumente wie Aufgabenstellung, Korrektur, etc. im öffentlichen Raum (Online, Aushang, Bekanntenkreis,...).
- Ich Sorge dafür, dass ich regelmäßige Updates meiner Lösungen in Uni2Work mache, sodass es zu verminderten technischen Problemen am Ende kommt. Die letzte innerhalb der Abgabefrist hochgeladene Version wird korrigiert. Achtung: Uni2Work logged Sie nach einer gewissen Zeit der Inaktivität automatisch wieder aus!

Die Prüfung besteht aus 1 Aufgaben. Die Punktezahl ist bei jeder Aufgabe angegeben.

Aufgabe	mögliche Punkte	erreichte Punkte
1.Allgemeine Fragen	18	
Summe:	18	
Note:		

Bepunktung Multiple Choice: Geben Sie an, welche Antworten zu folgenden Fragen zutreffen, indem Sie die Ziffern der richtigen Antworten notieren. Alle nicht notierten Ziffern sind automatisch als falsch klassifiziert (Sie können sich also nicht enthalten). Beispiel: Angenommen, die Antworten 1,2,3 sind korrekt, 4,5,6 sind falsch, so notieren Sie in der Antwort-Datei 1,2,3 (Antworten 4,5,6 werden *NICHT* notiert und sind damit automatisch als falsch gekennzeichnet).

Pro richtiger Antwort bekommen Sie einen halben Punkt. Die Punkte werden pro Aufgabenblock abgerundet.

Aufgabe 1 Allgemeine Fragen

(18 Punkte)

(a) Welche Aussagen über Komplexitäten sind korrekt?

- ☐ 1 $f(n) + 100n \in \mathcal{O}(f(n))$
- ☐ 2 $2^{n+a} \in \mathcal{O}(2^n)$ (mit a konstante positive Zahl).
- ☐ 3 Es sei $g(n) = \sum_{j=0}^n j$ und $f(n) = n^2$. Dann ist $g \in \mathcal{O}(f)$.
- ☐ 4 $f(n) \in \mathcal{O}(g(n)) \Leftrightarrow g(n) \in \mathcal{O}(f(n))$.
- ☐ 5 $f(n) \in \mathcal{O}(g(n)) \Rightarrow \forall n \in \mathbb{N} : f(n) < g(n)$.
- ☐ 6 $\mathcal{O}(n^2) \subset \mathcal{O}(2^n)$.

(b) Welche Aussagen über Sortieralgorithmen sind zutreffend?

- ☐ 1 QuickSort arbeitet im Worst-Case schneller als alle anderen Sortieralgorithmen.
- ☐ 2 SelectionSort wählt im k -ten Schritt das k -kleinste Element und tauscht mit dem Element bei k .
- ☐ 3 Mit HeapSort lassen sich nur Arrays sortieren, die die Heapeigenschaft besitzen.
- ☐ 4 MergeSort arbeitet nach dem Divide-and-Conquer-Prinzip.
- ☐ 5 BubbleSort kommt im besten Fall ohne Vertauschungsoperationen aus.
- ☐ 6 Terminierende und partiell korrekte Sortieralgorithmen heißen stabil.

(c) Welche Aussagen treffen auf Bäume zu?

- ☐ 1 Ein Binärbaum hat stets mindestens so viele Blätter wie innere Knoten.
- ☐ 2 Ein Binärbaum der Höhe h kann durch ein Array der Länge $2^{h+1} - 1$ repräsentiert werden.
- ☐ 3 In einem AVL-Baum der Höhe h gibt es mindestens zwei Pfade, die länger als $h - 2$ sind.
- ☐ 4 Bei gegebenen Schlüsselwerten ist die Anzahl Knoten n des zugehörigen B -Baums der Ordnung k eindeutig.
- ☐ 5 Ein B^+ -Baum der Ordnung k hat genau k Wurzeln.
- ☐ 6 Beim Einfügen in einen AVL-Baum kann sich nur die Balance von Knoten ändern, die auf dem Suchpfad liegen.

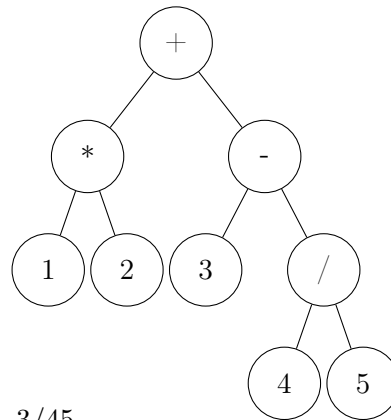
(d) Was gilt beim Suchen/Hashing?

- ☐ 1 Binäre Suche beschleunigt die lineare Suche durch paralleles Suchen von hinten nach vorne.
- ☐ 2 Beim linearen Sondieren wird jede Position der Hashtable getroffen.
- ☐ 3 Doppelhashing reduziert Clusterbildung.
- ☐ 4 Eine perfekte Hashfunktion kann beliebig viele Objekte kollisionsfrei einfügen.
- ☐ 5 Binäre Suche findet ein Element immer schneller als lineare Suche.
- ☐ 6 Beim Hashing mittels $h(x) = x \bmod 3$ erreicht quadratisches Sondieren einen Zykel nach 3 Schritten.

(e) Was trifft auf den minimal spannenden Baum $G' = (V', E')$ eines Graphen $G = (V, E)$ zu?

- ☐ 1 G' ist ein Subgraph mit gleichen Knoten $V' = V$, aber nur einer Teilmenge der Kanten $E' \subseteq E$.
- ☐ 2 Wenn zu G' noch eine beliebige Kante hinzugefügt werden würde, wäre G' kein Baum mehr.
- ☐ 3 In G' existiert mindestens ein Knoten mit Grad 1.
- ☐ 4 $|E'| + 1 = |V|$.
- ☐ 5 G' ist stets eindeutig.
- ☐ 6 Wenn man eine einzelne Kante aus G' löscht, so ist G' kein Baum mehr.

(f) Welche Graphtraversierungen passen zu folgendem Baum T ?



☐ 1 $Preorder(T) = + * 12 - 3/45.$

☐ 2 $Preorder(T) = + * 12 - /345.$

☐ 3 $Postorder(T) = 12345/ * -+.$

☐ 4 $Postorder(T) = 12 * 345/ -+.$

☐ 5 $Infixorder(T) = + * -123/45.$

☐ 6 $Infixorder(T) = 1 * 2 + 3 - 4/5.$