Name: Vorname: Matrikelnr.: Tutorium:

Jun.-Prof. Hofheinz, Jun.-Prof. Meyerhenke (ITI, KIT) 08.06.2015

Übungsklausur Algorithmen I

Aufgabe 1. (*Algorithm Engineering*) [2 Punkte]

Nennen Sie zwei Konzepte, die Algorithm Engineering im Gegensatz zu theoretischer Algorithmik ausmachen.

Aufgabe 2. (*Listen*) [4 Punkte]

- Nennen Sie zwei Operationen, die doppelt verkettete Listen in konstanter Worst-Case-Zeit unterstützen, einfach verkettete Listen jedoch nicht.
- Nennen Sie zwei Nachteile von verketteten Listen gegenüber beschränkten Feldern.

Aufgabe 3. (*Hashing mit Kollisionen*) [2 Punkte]

Nennen Sie einen Vorteil und einen Nachteil von Hashing mit verketteten Listen gegenüber Hashing mit linearer Suche.

Aufgabe 4. (*Dynamische Arrays*) [6 Punkte]

- Welche der folgenden Operationen eines unbeschränkten/dynamischen Arrays haben im schlimmsten Fall (ohne Amortisierung) $\Theta(n)$ Zeitkomplexität? pushBack, Folge von n pushBack, Elementzugriff, Abfrage der Größe
- Geben Sie die amortisierte Laufzeit der Operation pushBack an und zeigen Sie dies per amortisierter Analyse mit der Bankkontomethode. Nehmen Sie an, dass die Kapazität des Arrays verdoppelt wird, sobald sie nicht mehr ausreichend ist, und dass nur pushBack-Operationen ausgeführt werden.

Aufgabe 5. (*Sortieren*) [4 Punkte]

- Nennen Sie einen Vorteil von Radixsort gegenüber Mergesort (bei geeigneten Schlüsseln) und einen Vorteil von Mergesort gegenüber Quicksort.
- Warum wird Quicksort in der Praxis trotzdem sehr häufig eingesetzt?

Aufgabe 6. (*Lösen von Rekurrenzen*) [2 Punkte]

Bestimmen Sie die Lösungen der folgenden Rekurrenzen im Θ-Kalkül mit der einfachen Form des Master-Theorems:

$T(1) = 1$, $T(n) = 16T(n/4) + 4n$, $n = 4^k$, $k \in \mathbb{N}$
$S(1) = 55$, $S(n) = n + 3S(n/3)$, $n = 3^k$, $k \in \mathbb{N}$

Aufgabe 7. (O-*Kalkül*) [5 Punkte]

a. Zeigen Sie: $\log(n!) \in \mathrm{O}(n \log n)$. [2 Punkte]

b. Zeigen Sie: $\log(n!) \in \Omega(n \log n)$. [3 Punkte]

Aufgabe 8. (*Entwurf einer Datenstruktur*) [8 Punkte]

Eine Datenstruktur D soll Paare der Form $(Schluessel,\ Wert)$ speichern (sowohl *Schluessel* und *Wert* seien Zahlen aus $\mathbb{Z}$). Paare (x,c) und (y,c') mit $x = y$ dürfen in D **nicht gleichzeitig** vorkommen. Weiter soll D folgende Operationen mit jeweils gegebenem Laufzeitverhalten unterstützen (n bezeichne dabei die Anzahl der in D enthaltenen Paare):

- $insert(x : Schluessel,\ c : Wert)$
  fügt (x,c) in D ein. Ist schon ein Paar (y,c') mit $y = x$ in D vorhanden, so wird D nicht verändert. Der Zeitbedarf sei erwartet $\mathrm{O}(\log n)$.

- $removeMin : Schluessel \times Wert$
  entfernt aus D ein Paar (x,c) mit **minimalem Wert** c und liefert das entfernte Paar als Ergebnis zurück. Der Zeitbedarf sei erwartet $\mathrm{O}(\log n)$.

- $contains(x : Schluessel) : boolean$
  stellt fest, ob D ein Paar (y,c) mit $y = x$ enthält. Der Zeitbedarf sei erwartet $\mathrm{O}(1)$.

Anwendungsbedingt sei bekannt, dass in D zu jedem Zeitpunkt höchstens m Paare gleichzeitig vorhanden sind, d. h. es gilt stets $n \leq m$. Über die Beschaffenheit der auftretenden Paare wisse man im Voraus aber nichts.

a. Skizzieren Sie, wie Sie diese Datenstruktur und die drei beschriebenen Operationen realisieren würden. [6 Punkte]

b. Begründen Sie kurz, warum die drei Operationen in Ihrer Realisierung das geforderte Laufzeitverhalten aufweisen. [2 Punkte]

Aufgabe 9. (*Hashing*) [4 Punkte]

Betrachten Sie die folgende Hashtabelle:

0	1	2	3	4	5	6	7	8	9	10
05	01	27	37	44	34	19	71		99	

Zur Kollisionsauflösung wird Hashing mit linearer Suche (mit Puffer $m' = 1$) verwendet. Die Hashfunktion $h(x) = x$ DIV 10 bildet eine zweistellige Zahl $x \in \{01, \ldots, 99\}$ auf ihre höchstwertige Ziffer ab. Beispiel: $h(62) = 6$, $h(06) = 0$.

a. Sei die Tabelle in dem oben angegebenen Zustand. Geben Sie den Zustand der Tabelle nach dem Entfernen von 37 an. [2 Punkte]

0	1	2	3	4	5	6	7	8	9	10

b. Sei die Hashtabelle nun leer. Fügen Sie die Zahlen 12, 17, 88, 89, 05, 22, 48, 49, 50 in dieser Reihenfolge in die Hashtabelle ein. [2 Punkte]

0	1	2	3	4	5	6	7	8	9	10

Name: Vorname: Matrikelnr.: Tutorium: ID: 1

Bitte beachten Sie:

– Schreiben Sie auf alle Blätter Ihren Namen, Ihre Matrikelnummer und Ihre Tutoriumsnummer.

– Die Klausur enthält einschließlich diesem 6 Blätter.

– Es gibt 9 Aufgaben mit insgesamt 37 Punkten.

Aufgabe	1	2	3	4	5	6	7
Punkte							
Aufgabe	8	9					Σ
Punkte							