

Universität Leipzig Institut für Informatik Text Mining und Retrieval	Algorithmen und Datenstrukturen WS 2018/19 – Serie 5/6		
Jun.-Prof. Dr. Martin Potthast Dr. Jochen Tiepmar	Ausgabe am 19.12.2018	Abgabe am 16.01.2019	Seite 1/4

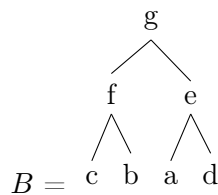
Algorithmen und Datenstrukturen – Serie 5/6

1 (10 Punkte) Heapsort

Gegeben sei ein Zahlenarray A mit 1-basiertem Index.

$$A = [1, 2, 5, 3, 6, 8, 4]$$

- a) Interpretieren Sie das Array als Darstellung eines Binärbaums und zeichnen Sie diesen. (2 Punkte)
- b) Stellen Sie mittels dem in der Vorlesung gegebenen Algorithmus *Build-Max-Heap* die Heap-Eigenschaft für den in a) erstellten Binärbaum her. Zeichnen Sie dazu den Baum nach jeder vollendeten Vertauschung. Achten Sie darauf, dass die Reihenfolge der erzeugten Bäume klar ersichtlich ist beziehungsweise nummerieren Sie sie gegebenenfalls. (4 Punkte)
- c) Gegeben sei Binärbaum B mit gegebener Heapeigenschaft.



Führen Sie mit B den Algorithmus *Heapsort* wie in der Vorlesung beschrieben durch. Verwenden Sie lexikographische Sortierung, also die Reihenfolge (von klein nach groß):

{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}

Zeichnen Sie für jeden Aufruf von *ExtractMax(A)* den Binärbaum vor und nach *MaxHeapify* und geben Sie den Rückgabewert an. Mögliche rekursive Aufrufe von *MaxHeapify* sollen nicht angegeben werden, können aber als Nebenrechnung hilfreich sein. Gehen Sie davon aus, dass Binärbäume automatisch in Arrayform übergeben werden können; Sie müssen also für die Anwendung des Pseudocodes nicht jeden Baum in ein Array transformieren. (4 Punkte)

Universität Leipzig Institut für Informatik Text Mining und Retrieval	Algorithmen und Datenstrukturen WS 2018/19 – Serie 5/6		
Jun.-Prof. Dr. Martin Potthast Dr. Jochen Tiepmar	Ausgabe am 19.12.2018	Abgabe am 16.01.2019	Seite 2/4

2 (10 Punkte) Quicksort

Gegeben sei ein Zahlenarray A mit 1-basiertem Index.

$$A = [10, 35, 43, 23, 47, 7, 15]$$

- a) Sortieren Sie dieses mit dem in der Vorlesung vorgestellten Algorithmus für Quicksort aufsteigend. Listen Sie den initialen und die rekursiven Funktionsaufrufe in folgender Tabelle auf. Geben Sie für jeden Aufruf der Funktion $qsort(A, p, r)$, bei dem $p < r$, die Parameter p , r und q , den betrachteten Inhalt von A zu Beginn des Aufrufs und die neu entstehenden rekursiven Aufrufe an. Ein (nicht zur Aufgabe passender) Aufruf $qsort(A, 2, 5)$ könnte also so beschrieben werden:

aktueller Aufruf	p	r	q	betr.Inh. v. A	neue Aufrufe
$qsort(A, 2, 5)$	2	5	4	1, 3, 31, 57, 47	$qsort(A, 2, 2)$, $qsort(A, 4, 5)$

Vergessen Sie keine rekursiven Aufrufe und listen Sie die Aufrufe in der Reihenfolge auf, in der sie ausgelöst werden. (7 Punkte)

- b) Wie muss ein Array mit fixer Länge (bspw. 7) im allgemeinen aufgebaut sein, dass Sortierung mit Quicksort maximale Schritte benötigt wenn als Pivotelement:
- i stets das letzte Element
 - ii stets ein zufälliges Element
 - iii stets das erste Element
- gewählt wird? (3 Punkte)

3 (3 Punkte) Bubblesort *Eigenrecherche*

Gegeben sei ein Zahlenarray A mit 1-basiertem Index.

$$A = [1, 2, 5, 3, 6, 4]$$

Sortieren Sie A mittels Bubblesort aufsteigend. Geben Sie nach jeder stattfindenden Vertauschung den vollständigen Arrayinhalt an.

4 (2 Punkte) Sortierparadigmen

Geben Sie für folgende Vorgänge an, welche Sortierparadigmen dabei umgesetzt werden.

- a) Ein Top-10 Ranking der besten Basketballspieler wird erstellt. Aufgrund von Speicherbeschränkung können dabei nicht alle vorsortiert werden.
- b) Beliebige Städtenamen werden einzeln zugerufen und direkt lexikographisch in eine sortierte Liste einsortiert.
- c) Mergesort.
- d) Bubblesort.

Universität Leipzig Institut für Informatik Text Mining und Retrieval	Algorithmen und Datenstrukturen WS 2018/19 – Serie 5/6		
Jun.-Prof. Dr. Martin Potthast Dr. Jochen Tiepmar	Ausgabe am 19.12.2018	Abgabe am 16.01.2019	Seite 3/4

5 (5 Punkte) Beweis durch Induktion

Gegeben sei die Funktion $f(n) = 3n + 2$.

- Beweisen Sie mittels vollständiger Induktion: $f(n) \in \Omega(n)$. Wählen Sie dazu $c = 1$ und $n_0 = 1$. (1 Punkt)
- Ist $f(n) \in \Theta(n)$? Begründen Sie kurz. Wenn ja, geben Sie ein passendes c an, sodass $f(n) \leq cn$. (1 Punkt)
- Ist $f(n) \in o(n)$? Begründen Sie kurz. (1 Punkt)
- Beweisen Sie mittels vollständiger Induktion: $f(n) \in O(n^2)$. Setzen Sie $c = 2$ und wählen Sie für n_0 die kleinstmögliche Zahl größer 0, sodass $f(n) \geq 2n^2$. (2 Punkte)

6 (7 Punkte) Stacks & Queues

Gegeben sei folgende kommaseparierte Auflistung von Elementen.

1, 2, 1, 2, 6, 0, 2

Geben Sie an, in welcher Reihenfolge die Elemente von den in a), b) und c) gegebenen Datenstrukturen bei der Leerung zurückgeliefert werden, nachdem diese in Auflistungsreihenfolge von links nach rechts eingefügt wurden. Bei der Priority Queue sollen alle Elemente mit der selben Priorität (bspw. 1) eingefügt werden. Wird ein bereits existierendes Element in die Priority Queue eingefügt, wird stattdessen die Priorität des existierenden Elements um 1 erhöht.

- Stack
- Queue
- Priority Queue
- Welche der Datenstrukturen aus a), b), c) bietet sich an, um Heapsort zu implementieren? Begründen Sie die Antwort mit ein bis zwei Sätzen. (2 Punkte)
- Geben Sie eine per Pseudocode beschriebene Funktion $printStack(S)$ an, die den Inhalt eines Stacks auf Konsole ausgibt. Die Ausgabe passiert über den Aufruf $print(x)$, die den Wert von x ausgibt. Achten Sie drauf, dass NIL kein auszugebender Wert ist. (2 Punkte)

7 (1 Punkt) Quizfrage

Wenn ein Supermarkt ein Algorithmus ist, dann sind Kassenschlangen:

- | | |
|------------|---------------------|
| (A) Stacks | (B) Schuppig |
| (C) Queues | (D) Priority Queues |

Universität Leipzig Institut für Informatik Text Mining und Retrieval	Algorithmen und Datenstrukturen WS 2018/19 – Serie 5/6		
Jun.-Prof. Dr. Martin Potthast Dr. Jochen Tiepmar	Ausgabe am 19.12.2018	Abgabe am 16.01.2019	Seite 4/4

8 (7 Punkte) Listen *Eigenrecherche*

Aufgrund eines irreparablen Datenfehlers ist eine doppelt verkettete Liste teilweise zerstört worden. Glücklicherweise stellt der Systemadministrator fest, dass kein Datenschlüssel verloren gegangen ist. Übrig geblieben sind folgende Anweisungen.

```
15.next = NIL
9.next = 15
8.prev = 6
4.prev = 8
```

Rekonstruieren Sie die doppelt verkettete Liste bestmöglich, indem Sie alle fehlenden Anweisungen auflisten. Listen Sie ebenfalls alle Elemente in Reihenfolge von *head* nach *tail* auf.

9 (5 Punkte) Hashing *Eigenrecherche*

Benutzen Sie Hashing, um die folgenden Anweisungen in der angegebenen Hashtabelle durchzuführen. Die Hashfunktion soll dabei den ersten Buchstaben jeder Zeichenkette zurückgeben. Die Variablenwerte sind durch Anführungszeichen umschlossen. Geben Sie die finale Hashtabelle an. Verwenden Sie Linear Probing zur Kollisionsbehandlung. Beschreiben Sie kurz den Ablauf der abschließenden Suchschritte. Überlaufbehandlung muss in der Aufgabe nicht bedacht werden. Wählen Sie selbstständig ein angemessenes Platzhaltersymbol.

```
Speichere "Hannelore"
Speichere "Gerhard"
Speichere "Ida"
Speichere "Justus"
Speichere "Immanuel"
Speichere "Johanna"
Entferne "Gerhard"
Entferne "Justus"
Suche "Johanna"
Suche "Gerhard"
```

Hashtabelle:

G	H	I	J	L	M
---	---	---	---	---	---