

Universität Leipzig Institut für Informatik Text Mining und Retrieval	<b>Algorithmen und Datenstrukturen</b> WS 2018/19 – Serie 3		
Jun.-Prof. Dr. Martin Potthast Dr. Jochen Tiepmar	Ausgabe am 21.11.2018	Abgabe am 28.11.2018	Seite 1/3

**Hinweis:** Die Übungsblätter werden soweit nicht anders angegeben vor Vorlesungsbeginn oder am Vortag 16 Uhr über den Briefkasten eingesammelt. Ausnahmen werden auf den Serien vermerkt und gelten nur für diese.

## Algorithmen und Datenstrukturen – Serie 3

### 1 (5 Punkte) Wert- und Referenzparameter

Gegeben sei der folgende Pseudocode bestehend aus den drei Funktionen *main*, *swap* und *addOne*.

```

main ()
    a = Array (2)
    a[1] = 5
    a[2] = 1
    print (a) //print1
    swap (a)
    print (a) //print2
    addOne (a)
    print (a) //print3

swap (a)
    tmp = a[1]
    a[1] = a[2]
    a[2] = tmp

addOne (a)
    a[1] = a[1]+1
    a[2]++ //entspricht a[2]=a[2]+1

```

Die *print*-Funktion gibt den Arrayinhalt kommasepariert in Indexreihenfolge aus und wurde zur besseren Übersicht durchnummeriert. **Entgegen den Angaben in der Vorlesung können Arrays hier auch als Wertparameter behandelt werden.** Geben Sie an, was in den folgenden Fällen durch die Ausführung von *main()* von allen drei *print*-Funktionen ausgegeben würde, wenn *swap* und *addOne* jeweils als *call by value* (*V*) oder *call by reference* (*R*) implementiert würden.

	swap	addOne	print1	print2	print3
a)	R	V			
b)	V	R			
c)	V	V			
d)	R	R			

Universität Leipzig Institut für Informatik Text Mining und Retrieval	<b>Algorithmen und Datenstrukturen</b> WS 2018/19 – Serie 3		
Jun.-Prof. Dr. Martin Potthast Dr. Jochen Tiepmar	Ausgabe am 21.11.2018	Abgabe am 28.11.2018	Seite 2/3

## 2 (10 Punkte) Mergesort

Gegeben sei ein Zahlenarray  $A$  mit 1-basiertem Index.

$$A = [9, 8, 7, 6, 1, 2, 3, 6, 5, 42]$$

- a) Sortieren Sie das Array  $A$  aufsteigend mit rekursivem Mergesort. Folgen sie genau dem in der Vorlesung gegebenen Algorithmus. Geben Sie das Ergebnis über folgende Tabelle an, wobei jeder vollständige Aufruf von  $\text{MergeSort}(A, p, r)$  eine Zeile bilden. (3 Punkte)

**Hinweis: Das Array muss nicht angegeben werden, aber es kann helfen, eines als Nebenrechnung mitzuführen. Sie dürfen in dieser Aufgabe *MergeSort* mit *MS* abkürzen.**

Aufruf	q	p	r	neue Aufrufe
--------	---	---	---	--------------

- b) Zeichnen Sie den Rekursionsbaum für den Aufruf  $\text{MergeSort}(A, 1, 5)$  aus a) analog zu der in der Vorlesung verwendeten Darstellung der rekursiven Berechnung der Fibonacci-Folge. Verwenden Sie die Funktionsaufrufe als Bezeichner für die Knoten (Das oberste  $F(5)$  der Vorlesungsdarstellung entspricht also hier  $MS(A, 1, 5)$ ). (3 Punkte)
- c) Welche Änderungen an der in a) erstellten Tabelle ergeben sich, wenn stattdessen das gespiegelte Array  $A' = [42, 5, 6, 3, 2, 1, 6, 7, 8, 9]$  als Input verwendet wird? (1 Punkt)
- d) Wie muss ein Array aufgebaut sein, um die minimale sowie maximale Menge an rekursiven Funktionsaufrufen für rekursiven Mergesort zu benötigen? (1 Punkt)
- e) Recherchieren Sie selbstständig den iterativen nicht-rekursiven Algorithmus für Mergesort und führen Sie ihn für  $A$  durch. Geben Sie jeweils für jede Granularitätsstufe den Arrayinhalt vor dem merge-Vorgang - also üblicherweise zum Anfang eines äusseren Schleifendurchlaufs - an, sowie das Endergebnis. (1 Punkt)
- f) Mit welchem sich auf natürliche Weise anbietenden Trick kann der Algorithmus aus e) verbessert werden? (1 Punkt)

Universität Leipzig Institut für Informatik Text Mining und Retrieval	<b>Algorithmen und Datenstrukturen</b> WS 2018/19 – Serie 3		
Jun.-Prof. Dr. Martin Potthast Dr. Jochen Tiepmar	Ausgabe am 21.11.2018	Abgabe am 28.11.2018	Seite 3/3

### 3 (10 Punkte) Memoisation

- a) Berechnen Sie den Fibonacci-Wert von 6 mit dem in der Vorlesung vorgestellten Algorithmus *Recursive Fibonacci*. Geben Sie das Ergebnis als Tabelle an, bei der jeder Aufruf von  $\text{RFib}(n)$  eine Zeile ist. Geben Sie die Aufrufe in der Reihenfolge an, in der sie ausgelöst werden. Nehmen Sie an, dass beim Operator  $+$  der linke Operand vor dem rechten Operand ausgewertet wird, also bspw. bei der Operation  $A + B$  der Operand  $A$  vor  $B$  ausgewertet wird. Geben Sie für jede Zeile den **aktuellen Aufruf** sowie **alle** durch diesen direkt **neu erzeugten Funktionsaufrufe** und den **RETURN-Wert** des jeweiligen Aufrufs als Tabellenspalten an. (6 Punkte)

aktueller Aufruf	neu erzeugte Aufrufe	RETURN
------------------	----------------------	--------

- b) Berechnen Sie den Fibonacci-Wert von 6 mit dem in der Vorlesung vorgestellten Algorithmus *Memoized Recursive Fibonacci*. Geben Sie das Ergebnis analog zu a) an. Verwenden Sie auch dieselbe Operanden-Reihenfolge. Geben Sie als zusätzliche Spalte den Inhalt von  $A$  nach Durchlauf des jeweiligen Aufrufs an. (4 Punkte)

aktueller Aufruf	neu erzeugte Aufrufe	RETURN	A
------------------	----------------------	--------	---