

Deckblatt | 5. Juli 2022

Name:	
Vorname:	
Semester:	
Matrikelnummer:	

**Hinweise:**

- Bitte legen Sie Ihren Studierendenausweis bereit.
  - Notieren Sie auf jedem Blatt Name und Matrikelnummer.
  - Bitte lesen Sie die Aufgabenstellungen genau durch.
  - Erlaubte Hilfsmittel: alle nicht-programmierbaren Hilfsmittel (Skripte, Übungsblätter etc.)
  - Bitte keinen Rotstift verwenden.
  - Die Prüfungsdauer beträgt 120 Minuten.
  - Es können insgesamt 120 Punkte erreicht werden.
- Viel Erfolg!

Aufgabe	Thema	max. Punkte	erreichte Punkte
1	Binäre Suchbäume	20	
2	Laufzeit / Komplexität	10	
3	Algorithmen	30	
4	Greedy-Verfahren	30	
5	Wissensfragen	10	
6	Heaps	20	
7			
Gesamt		120	

### Aufgabe 1: Binäre Suchbäume (20 Punkte)

- a) Erzeugen Sie aus den folgenden Eingabereihenfolgen je einen binären Suchbaum nach dem in der Vorlesung verwendeten Algorithmus und zeichnen Sie diese Bäume.

1) 12, 14, 17, 10, 25, 11, 20, 6, 26

2) 26, 19, 15, 24, 21, 22, 28, 18, 20

3) 31, 17, 14, 16, 12, 3, 1, 2, 24

- b) Löschen Sie aus dem Baum aus a.2) zunächst den Knoten mit Schlüssel 26, danach den mit Schlüssel 19 und zuletzt denjenigen mit Schlüssel 24. Verwenden Sie den in der Vorlesung besprochenen Algorithmus. Zeichnen Sie die drei Bäume.

### Aufgabe 2: Laufzeit / Komplexität (10 Punkte)

Bestimmen Sie die Laufzeiten von

a)  $T(n) = 4T\left(\frac{n}{4}\right) + 5$

b)  $T(n) = 4T\left(\frac{n}{2}\right) + 2n^2$

### Aufgabe 3: Algorithmen (30 Punkte)

Betrachten Sie den folgenden Algorithmus. Eingabeparameter ist ein Array  $A$ , Rückgabewert ist ein boolescher Wert (True oder False).

```
1: function CHECK( $A$ )
2:    $n$  = Länge des Arrays  $A$ 
3:   resultat = False
4:   for  $i = 1$  bis  $n$  mit Schrittweite 1 do
5:     for  $j = n$  bis  $i + 1$  mit Schrittweite  $-1$  do
6:       if  $A[j] == A[i]$  then
7:         resultat = True
8:       end if
9:     end for
10:  end for
11:  return resultat
12: end function
```

- a) (6) Was bewirkt der Algorithmus? Testen Sie ihn an den beiden Eingabearrays:
- 1) 5, 7, 1, 8, 3, 4
  - 2) 5, 7, 1, 8, 5, 7
- b) (9) Welches ist die Komplexitätsklasse des Algorithmus?
- c) (3) Was gilt für die Array-Elemente beim Worst-case des Algorithmus?
- d) (3) Was gilt für die Array-Elemente beim Best-case des Algorithmus?
- e) (9) Angenommen, alle Array-Elemente haben den gleichen Wert. Geben Sie eine Formel an, die in Abhängigkeit von der Länge  $n$  des Arrays angibt, wie oft Zeile 7 ausgeführt wird. Zählen Sie die Anzahl der Ausführungen für Arrays der Länge 1, 2, 3 und 4 und überlegen Sie sich dann eine allgemeine Formel.

#### Aufgabe 4: Greedy-Verfahren (30 Punkte)

Ein Backup-System muss 10 verschieden große Backup-Dateien auf 5 Festplatten der Größe 1 TB = 1024 GB verteilt abspeichern. Sie dürfen davon ausgehen, dass die 1024 GB komplett verwendet werden können. Die Festplatten sind nummeriert. Hierzu wird ein Greedy-Algorithmus verwendet, der die Dateien speichert.

Es gibt eine Menge von Backup-Dateien, deren Größe  $s$  und Aktualität  $a$  in folgender Tabelle gespeichert sind (Größe angegeben in GB, Aktualität = wie viele Tage liegt das Backup zurück)

$i$	1	2	3	4	5	6	7	8	9	10
$s_i$	820	600	430	1015	190	360	420	220	800	40
$a_i$	2	6	4	0	9	3	7	1	8	5

- Entwerfen Sie einen Algorithmus nach dem Greedy-Prinzip, der die Backup-Dateien so verteilt, dass er zuerst die neuesten Dateien abspeichert und dazu immer die erste Festplatte (sortiert nach der Nummerierung der Festplatten) sucht, auf welcher die Datei gesichert werden kann.
- Wie sieht die Verteilung der Backup-Dateien mit Ihrem Algorithmus auf den einzelnen Festplatten aus? Am einfachsten lässt sich das tabellarisch
- Wie sieht die Verteilung der Dateien aus, wenn die Dateien absteigend nach ihrer Größe ausgewählt werden? Am einfachsten lässt sich das tabellarisch darstellen. Werden alle Dateien abgespeichert?
- Wie sieht die Verteilung der Dateien aus, wenn die Dateien aufsteigend nach ihrer Größe ausgewählt werden? Am einfachsten lässt sich das tabellarisch darstellen. Werden alle Dateien abgespeichert?

### Aufgabe 5: Wissensfragen (10 Punkte)

a) Das Speicherprinzip eines Stacks heißt

- ☐ FIFO    ☐ FILO    ☐ LIFO

b) Ein absteigend sortiertes Array repräsentiert einen Max-Heap.

- ☐ wahr    ☐ falsch

c) Im Fall eines bereits aufsteigend sortierten Arrays mit N Elementen benötigt der Partitionierungsalgorithmus bei QuickSort

- ☐ Keine Vertauschungen  
☐ Eine Vertauschung jedes Elementes mit sich selbst  
☐ Genau  $N/2$  Vertauschungen

d) MergeSort arbeitet nach dem Verfahren „Teile und Beherrsche“.

- ☐ wahr    ☐ falsch

e) Ein rekursiver Algorithmus kann optimiert werden durch folgendes Algorithmen-Verfahren:

### Aufgabe 6: Heaps (20 Punkte)

Gegeben ist das Array **{ 50, 30, 5, 20, 10, 40 }**

Dieses ist mit dem Algorithmus HEAP-SORT aus der Vorlesung zu sortieren. Geben Sie den Inhalt des Arrays nach dem Aufruf von BUILD-MAX-HEAP sowie nach jedem Aufruf von MAX-HEAPIFY an. Tragen Sie Ihre Ergebnisse in Feldern ein wie das, das im Folgenden abgebildet ist.

50	30	5	20	10	40
----	----	---	----	----	----