

Klausur „Algorithmen I“

Name
Vorname
Matrikelnummer
Semester

- Das erste Blatt der abgegebenen Klausur muss dieses **Deckblatt** sein.
- Jede Aufgabe beginnt auf einem **neuen Blatt**.
- Jedes Blatt ist nur auf der **Vorderseite** beschrieben.
- Die Blätter sind nach Aufgaben **aufsteigend sortiert**.
- Die Blätter sind **zusammengeheftet**. Nicht angeheftete Blätter werden nicht gewertet.

Aufgabe	1	2	3	4	5	6
Max. Punktzahl	20	20	10	15	20	15
Punkte						

Aufgabe 1

20 Punkte

Sie haben einen Rucksack, der ein Gewicht von 60 kg tragen kann, sowie eine Menge von Gegenständen $i = 1, \dots, 8$ mit Gewichten g_i (in kg) sowie Werten w_i , die Sie aus der folgenden Tabelle entnehmen können:

i	1	2	3	4	5	6	7	8
g_i	10	12	6	5	20	13	17	16
w_i	3	4	2	5	8	6	9	7

1. Nehmen Sie an, Sie wollen möglichst viele Gegenstände in den Rucksack packen:
 - a) Welches Kriterium können Sie für einen nach dem Greedy-Prinzip arbeitenden Algorithmus wählen, um Ihr Ziel zu erreichen?
 - b) In welcher Reihenfolge packt ein solcher Algorithmus die Gegenstände in den Rucksack?
 - c) Wie schwer ist der Rucksack am Ende und welchen Wert haben die Gegenstände in der Summe?
2. Nehmen Sie an, Sie wollen Gegenstände mit einem möglichst hohen Gesamtwert in den Rucksack packen:
 - a) Welches Kriterium können Sie für einen nach dem Greedy-Prinzip arbeitenden Algorithmus wählen, um Ihr Ziel zu erreichen?
 - b) In welcher Reihenfolge packt ein solcher Algorithmus die Gegenstände in den Rucksack?
 - c) Wie schwer ist der Rucksack am Ende und welchen Wert haben die Gegenstände in der Summe?

Aufgabe 2

20 Punkte

1. Sortieren Sie das Array $A = [8, 5, 1, 9, 7, 3, 2]$ mit dem in der Vorlesung besprochenen Insertionsort-Algorithmus. Geben Sie das Array nach jedem Durchlauf der äußersten Schleife an.
2. Sortieren Sie das Array $A = [7, 6, 5, 9, 4, 8, 2, 3]$ mit dem in der Vorlesung besprochenen Quicksort-Algorithmus. Geben Sie das Array nach jeder Partitionierungs-Operation an.

Aufgabe 3

10 Punkte

Bestimmen Sie mithilfe des Mastertheorems die Komplexitätsklasse der Rekursionsgleichung

$$T(n) = 3 \cdot T\left(\frac{n}{27}\right) + \sqrt[3]{n+1}$$

Hinweis: Seien Sie sorgfältig bei der Bestimmung von $\log_b a$. Denken Sie lieber zweimal nach.

Aufgabe 4

15 Punkte

Gegeben sei folgender Algorithmus:

```
1: function MAGIC(x, n)
2:   if n == 0 then
3:     return 1
4:   end if
5:   if n ist gerade then
6:     x2 = MAGIC(x, n/2)
7:     return x2*x2
8:   else
9:     x2 = MAGIC(x, (n-1)/2)
10:    return x2*x2*x
11:  end if
12: end function
```

1. Testen Sie die Funktionsweise des Algorithmus durch Aufruf von

- a) `magic(4, 2)`
- b) `magic(4, 3)`
- c) `magic(4, 4)`
- d) `magic(4, 5)`

Geben Sie bei jedem rekursiven Aufruf die Werte der Parameter an. Was berechnet der Algorithmus?

2. Schreiben Sie eine Rekursionsgleichung für die Rechenzeitkomplexität des Algorithmus auf. Sie müssen die Rekursionsgleichung nicht lösen.

Aufgabe 5

20 Punkte

1. Erzeugen Sie aus den folgenden Eingabereihenfolgen je einen binären Suchbaum nach dem in der Vorlesung verwendeten Algorithmus und zeichnen Sie diese Bäume
 - a) 45, 92, 12, 9, 89, 4, 25, 38, 1, 2
 - b) 9, 4, 1, 8, 5, 2, 7, 6, 3
 - c) 74, 35, 22, 23, 21, 6, 4, 1
2. Löschen Sie aus dem Baum aus 1.b) zunächst den Knoten mit Schlüssel 9, danach den mit Schlüssel 1 und zuletzt denjenigen mit Schlüssel 4. Verwenden sie den in der Vorlesung besprochenen Algorithmus. Zeichnen Sie die drei Bäume.

Aufgabe 6

15 Punkte

Erzeugen Sie durch Eingabe von 6, 4, 8, 3, 1 einen Rot-Schwarz-Baum. Zeichnen Sie die 5 entstehenden Bäume.

Gesamtpunktzahl: 100

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--

--	--	--	--	--	--	--	--
