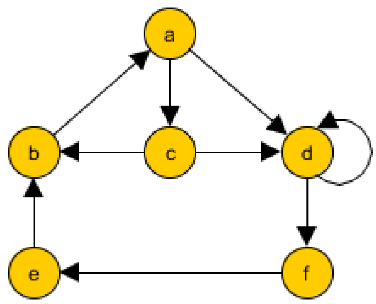
Name:________________________________ Matr.-Nr.:___________________

Aufgabe 1 *(10 Punkte)*



Gegeben sei nebenstehender Graph G:

a) Ist G schwach zusammenhängend? Ist G stark zusammenhängend? Begründen Sie Ihre Antworten jeweils. *(2 Punkte)*

b) Welche der folgenden Kanten könnten in G entfernt werden, ohne dass dies die Zusammenhangskomponenten von G verändert? *(2 Punkte)*

Kante	Könnte entfernt werden (j/n)
(d, d)	
(c, b)	
(a, c)	
(a, d)	

c) Nennen Sie vier Grapheneigenschaften von G, die in Aufgabenteil a) nicht bereits diskutiert wurden. (*2 Punkte*)

d) Überprüfen Sie, ob der Graph Zyklen der in der Tabelle angegebenen Länge enthält. Falls ja, geben Sie jeweils ein Beispiel an. *(4 Punkte)*

Zyklenlänge	Beispiel
1	
2	
3	
4	
5	
6	
7	
8	

Aufgabe 2 *(10 Punkte)*

Gegeben sei die Adjazenzmatrix A eines gerichteten Graphen G=(V, E). A ist ein zweidimensionales, quadratisches Array A[0..n-1][0..n-1] mit n Zeilen und Spalten. Alle Elemente im Array stammen aus der Menge {0, 1}. Ferner gelte $n \geq 1$. Geben Sie einen **vollständigen** Algorithmus **in Pseudocode-Notation** an, der als einzige Eingabe A erhält und ermittelt, ob für über die Hälfte der Knotenpaare (x, y) mit $x \neq y$ sowohl eine Kante von x nach y als auch von y nach x existiert.

Hinweise:

- (x, y) und (y, x) sind als ein einziges Knotenpaar zu betrachten.
- Bei dieser Aufgabe kommt es nur auf die Korrektheit des Algorithmus und der Nutzung der Pseudocode-Notation an, nicht auf die Effizienz des Algorithmus.

Aufgabe 3 *(10 Punkte)*

a) Gegeben sei untenstehender Algorithmus f(m, n). Bestimmen Sie für den Algorithmus die **genaue Laufzeitfunktion** und die **Komplexitätsklasse**. Verwenden Sie als Basisoperation die **Addition**. Gehen Sie dabei auch auf **Best- und Worst-Case** ein.
(7 Punkte)
Hinweis: Die Funktion floor(x) liefert die größte Ganzzahl, die kleiner oder gleich x ist.

```
ALGORITHM f(m, n)
    // Input: zwei nicht-negative ganze Zahlen m und n
    result ← 1.0
    i ← n
    while i > 1 do
        j ← 1
        while j * j ≤ m do
            result ← result + (1/result)
            j ← j + 2
        i ← floor(i/3)
    return result
```

b) Zur Lösung eines Problems stehen zwei Algorithmen *$A_1(n)$* und *$A_2(n)$* zur Auswahl. Dabei repräsentiert n die Anzahl der Kunden eines Kleinunternehmens. Für die Algorithmen gilt jeweils: *best case = worst case*.
Die Laufzeitfunktionen der Algorithmen sind $T_1(n) = \frac{1}{5}n^2$ und $T_2(n) = 8n \cdot \sqrt{n}$. Welcher Algorithmus sollte bei welcher Kundenzahl aus Effizienzgründen verwendet werden? Begründen Sie Ihre Antwort kurz und geben Sie die relevanten Rechenschritte an.
(3 Punkte)

Aufgabe 4 *(10 Punkte)*

a) Gegeben sei untenstehender Algorithmus f(n, a). Stellen Sie für f(n, a) die **Rekursionsgleichung** der Laufzeitfunktion T(n) auf. Verwenden Sie als Basisoperation die **Addition**. Bestimmen Sie die **genaue Laufzeitfunktion** in geschlossener (d. h. nicht-rekursiver) Schreibweise durch **rückwärtiges Einsetzen**. (Ein Beweis mittels vollständiger Induktion ist nicht notwendig.) Geben Sie außerdem die **Komplexitätsklasse** des Algorithmus an. *(8 Punkte)*

Hinweise:

- Die Funktion floor(x) liefert die größte Ganzzahl, die kleiner oder gleich x ist.

```
ALGORITHM f(n, a)
    // Input: Zwei nicht-negative ganze Zahl n und a
    if n > 2 then
        res ← f(floor(n/3), a+1)
        res ← res * f(floor(n/3), a+2)
        res ← res * f(floor(n/3), a+3)
        return res
    else
        return a+1
```

b) Erläutern Sie in Ihren eigenen Worten, welche **Aufgabe** die **Konstanten c_1 und c_2** in der Definition der **θ-Notation** besitzt. Gehen Sie dabei insbesondere auf die Motivation für die Einführung der Konstanten ein. *(2 Punkte)*

Aufgabe 5 *(10 Punkte)*
Gegeben sei folgendes Rucksackproblem:

Rucksackkapazität W = 6 kg

Gegenstand	Gewicht	Wert
A	1 kg	90 Euro
B	4 kg	300 Euro
C	3 kg	240 Euro
D	2 kg	160 Euro

Lösen Sie das Problem mit Hilfe der Dynamischen Programmierung:

- Erstellen und beschriften Sie die Tabelle V(i, j).
- Füllen Sie anschließend die Tabelle aus.
- Geben Sie die Menge der eingepackten Gegenstände an und kennzeichnen Sie, wie diese Gegenstände mithilfe der Tabelle bestimmt werden können.
- Wie hoch ist der Gesamtwert der eingepackten Gegenstände?

Aufgabe 6 *(10 Punkte)*

Gegeben sei folgendes Array von Werten [k, u, c, h, e, n]. Sortieren Sie die Werte mit Hilfe der unten genannten Sortierverfahren.

a) **Selectionsort** in **aufsteigender** alphabetischer Reihenfolge: Geben Sie den Zustand des Arrays **nach jedem Durchlauf der äußeren Schleife** an. *(3 Punkte)*

b) **Heapsort** in **absteigender** alphabetischer Reihenfolge: Stellen Sie die Aufbau- und Entnahmephase **detailliert in Array-Form** dar.
(7 Punkte)

Aufgabe 7 *(10 Punkte)*

Gegeben sei der folgende Text „KREPPPAPIER“

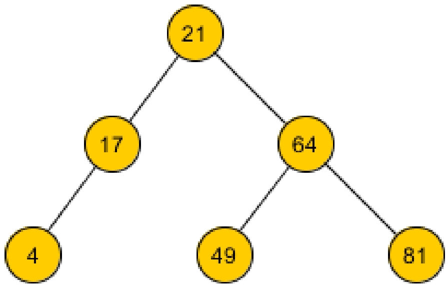
a) Bestimmen Sie die relative Häufigkeit aller auftretenden Buchstaben. *(2 Punkte)*

b) Bestimmen Sie eine möglichst effiziente binäre Codierung der Buchstaben mit Hilfe des Huffman-Algorithmus. Zeichnen Sie dazu für jeden Durchlauf die entsprechenden Binärbäume. *(5 Punkte)*

c) Wie viele Bits werden zur Codierung des Textes benötigt? Wie viele Bits würden bei fixer Codierung des Alphabets entstehen? Skizzieren Sie jeweils Ihren Rechenweg. *(3 Punkte)*

Aufgabe 8 *(10 Punkte)*

a) Gegeben sei folgender AVL-Baum:



Fügen Sie nacheinander die Schlüssel 31, 27 und 42 ein. Zeichnen Sie den Baum **nach jeder Einfügeoperation**. Falls Sie beim Einfügen rotieren müssen, zeichnen Sie den Baum auch **vor jeder einzelnen Rotation**. Geben Sie ferner jeweils an, **welche Art von Rotation** Sie **um welchen Knoten** ausführen. *(6 Punkte)*

b) Wie viele Schlüssel kann ein **2-3-Baum** mit der **Höhe 4 höchstens** enthalten? Wie viele muss er **mindestens** besitzen? Erläutern Sie jeweils Ihren **Rechenweg**. *(4 Punkte)*

Name:________________________________ Matr.-Nr.:_________________

Aufgabe 9 *(10 Punkte)*

Gegeben seien das Alphabet {A, C, D, E, H, I, M, N, O, Ö, R, S, Leerzeichen}, der Text „DER MONDSCHEIN SCHIEN SCHON SCHÖN“ und das Suchmuster „SCHÖN“.

a) Erstellen Sie die entsprechende Shift-Tabelle des Horspool-Algorithmus. *(2 Punkte)*

A	C	D	E	H	I	M	N	O	Ö	R	S	_

b) Wie sucht der Horspool-Algorithmus das gegebene Suchmuster im Text? Zeichnen Sie dazu die Zeichenvergleiche und Verschiebungen auf. *(4 Punkte)*

D	E	R	_	M	O	N	D	S	C	H	E	I	N	_	S	C	H	I	E	N	_	S	C	H	O	N	_	S	C	H	Ö	N

c) Wie viele **Verschiebungen** und **Zeichenvergleiche** werden bei der Suche im gegebenen Beispiel durch den **Horspool-Algorithmus** ausgeführt? Wie sehen die Werte bei der **Brute Force-Methode** aus? **Erläutern** Sie jeweils Ihren **Rechenweg** bzw. **markieren** Sie aufgetretenen Verschiebungen/Vergleiche. *(4 Punkte)*

Aufgabe 10 *(10 Punkte)*

Kreuzen Sie an, ob folgende Aussagen entweder wahr oder falsch sind. (*Hinweis*: Inkorrekte Antworten führen **nicht** zu Abzügen.):

Aussage	*wahr*	*falsch*
1. Es gibt Algorithmen, die bei gleicher Eingabe immer die gleiche Ausgabe produzieren, obwohl diese Algorithmen nicht deterministisch sind.		
2. Quicksort eignet sich besonders gut für das Sortieren von verketteten Listen.		
3. Eine Hash-Tabelle speichert die enthaltenen Elemente in sortierter Reihenfolge.		
4. 2-3-Bäume eignen sich besonders gut für die Suche in großen Datenmengen auf externen Speichermedien.		
5. Nicht für jedes Problem existieren Algorithmen, die in einer besseren Komplexitätsklasse liegen als die Brute-Force-Lösung des Problems.		
6. Auf Basis der Tiefensuche kann jeder zyklenfreie Graph topologisch sortiert werden.		
7. Mit Hilfe des Master-Theorems lässt sich für bestimmte rekursive Algorithmen die Komplexitätsklasse ermitteln, nicht aber die genaue Laufzeitfunktion.		
8. Die Breitensuche kann sowohl bei gerichteten als auch ungerichteten Graphen genutzt werden.		
9. Ein Einstiegspunkt eines DAG ist ein Knoten, von dem keine Kante ausgeht.		
10. Ein Knoten eines B-Baums der Ordnung m kann bis zu m Schlüssel enthalten.		

Zusätzlicher Platz zur Aufgabenbearbeitung

Zusätzlicher Platz zur Aufgabenbearbeitung