Student/in:		Unterschrift:	
DHBW Duale Hochschule Baden-Württemberg Stuttgart **ÜBUNGSKLAUSUR**		Fakultät Studiengang: Jahrgang / Kurs : Studienhalbjahr:	**Technik** **Angewandte Informatik** **2014 B/C/D/K** **2. Semester**
Datum:	**16/17. Juli 2015**	Bearbeitungszeit:	**90 Minuten**
Modul: Unit:	**T2INF1003.1** **Algorithmen**	Dozent:	**Jan Hladik** **Stephan Schulz**
Hilfsmittel:	**Vorlesungsskript, eigene Notizen**		
Punkte:		Note:	

Aufgabe	erreichbar	erreicht
1	9	
2	9	
3	9	
4	8	
5	6	
6	9	
7	11	
Summe	61	

1. Sind Sie gesund und prüfungsfähig?
2. Sind Ihre Taschen und sämtliche Unterlagen, insbesondere alle nicht erlaubten Hilfsmittel, seitlich an der Wand zum Gang hin abgestellt und nicht in Reichweite des Arbeitsplatzes?
3. Haben Sie auch außerhalb des Klausurraumes im Gebäude keine unerlaubten Hilfsmittel oder ähnliche Unterlagen liegen lassen?
4. Haben Sie Ihr Handy ausgeschaltet und abgegeben?

(Falls Ziff. 2 oder 3 nicht erfüllt sind, liegt ein Täuschungsversuch vor, der die Note „nicht ausreichend“ zur Folge hat.)

Aufgabe 1 (4+2+3 Punkte)
Betrachten Sie die Folge
$$S = \{3, 8, 17, 5, 15, 2, 1, 12, 4, 9\}$$

a) Sortieren Sie die Folge S mit dem Selection-Sort-Verfahren. Geben Sie hierzu den Zustand von S nach jeder Vertauschungsoperation an.

b) Wie viele Vertauschungen benötigen Sie?

c) Wie viele Vergleiche von Elementen aus S benötigen Sie?

Aufgabe 2 (3+3+3 Punkte)

a) Betrachten Sie die Funktion $f : \mathbb{N} \to \mathbb{R}, f(x) = 10 - \frac{3}{x}$. Zeigen oder widerlegen sie: $f \in \mathcal{O}(1)$

b) Betrachten Sie die Funktion $g : \mathbb{N} \to \mathbb{R}, g(x) = 2x - \frac{4}{x^2}$. Zeigen oder widerlegen sie: $g \in \Theta(x)$

c) Zeigen Sie: $x^2 + \log_2 x \in \mathcal{O}(x^3)$

Fortsetzung

Aufgabe 3 (1+1+1+6 Punkte)
Betrachten Sie die folgende C-Funktion:

```
int machwas2(int n)
{
    int i,j, res;
    int k = n;
    res = 0;
    for(i=0; i< n; i++)
    {
        for(j=0; j<k; j++)
        {
            res=res+n;
        }
        k = k/3;
    }
    return res;
}
```

a) Bestimmmen Sie den Rückgabwert für die Eingaben $n = 4, n = 8, n = 12$.

b) Bestimmen Sie das kleinste $k \in \mathbb{N}$ so dass die Laufzeitkomplexität von `machwas2()` in $\mathcal{O}(n^k)$ ist. Begründen Sie ihre Behauptung!

Aufgabe 4 (3+3+2 Punkte)
Betrachten Sie die Folge

$$S = \{3, 8, 17, 5, 15, 2, 1, 12, 4, 9\}$$

a) Partitionieren Sie (einmal) die Folge S mit dem in der Vorlesung unter dem Namen `q_part()` beschriebenen LL-Partitions-Algorithmus. Verwenden Sie dabei aber jeweils das *erste* Element einer (Teil-) Folge als Pivot. An welcher Stelle steht nach der Partitionierung das Pivot-Element?

b) Sortieren Sie die Folge S mit dem in der Vorlesung beschriebenen LL-Quicksort-Algorithmus zu Ende. Verwenden Sie weiterhin jeweils das *erste* Element einer (Teil-) Folge als Pivot.

c) Wie viele Vergleiche von Elementen benötigen Sie insgesamt (also in Teilen a und b)?

Aufgabe 5 (3+3 Punkte)

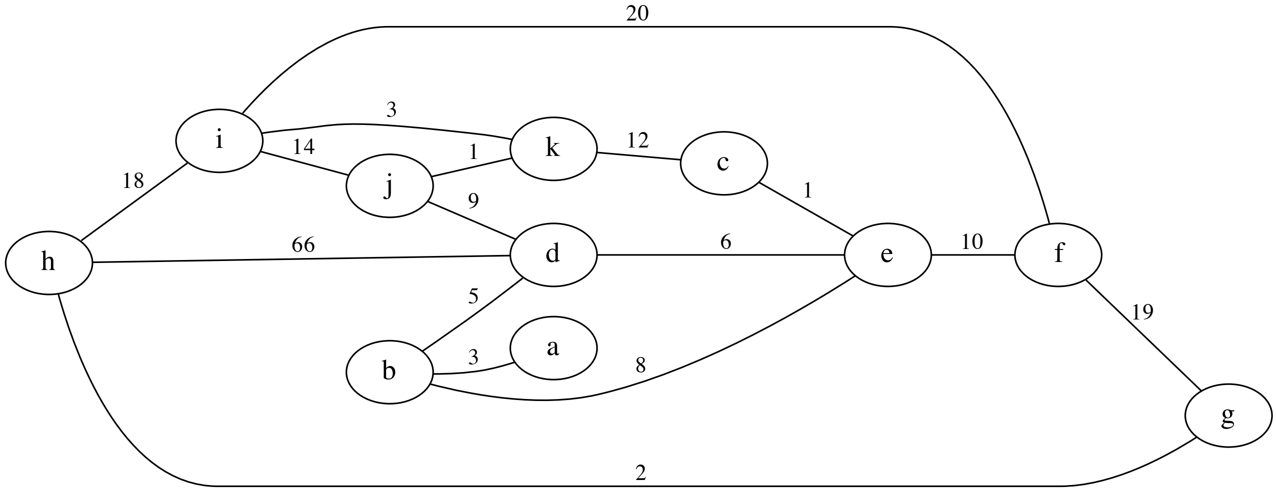
a) Betrachte Sie folgende Rekurrenzrelation und lösen Sie diese (mindestens durch Angabe einer guten $\mathcal{O}()$-Schranke). Sie können davon ausgehen, dass $R(0) = 0$ gilt:

$$R(n) = 4 \cdot R(\frac{n}{2}) + 4n^3 + n$$

b) Betrachte Sie folgende Rekurrenzrelation und lösen Sie diese (mindestens durch Angabe einer guten $\mathcal{O}()$-Schranke). Sie können davon ausgehen, dass $S(0) = 0$ gilt:

$$S(n) = 2 \cdot S(n-1) + 1$$

Aufgabe 6 (3+2+4 Punkte)
Gegeben sei der Graph G:

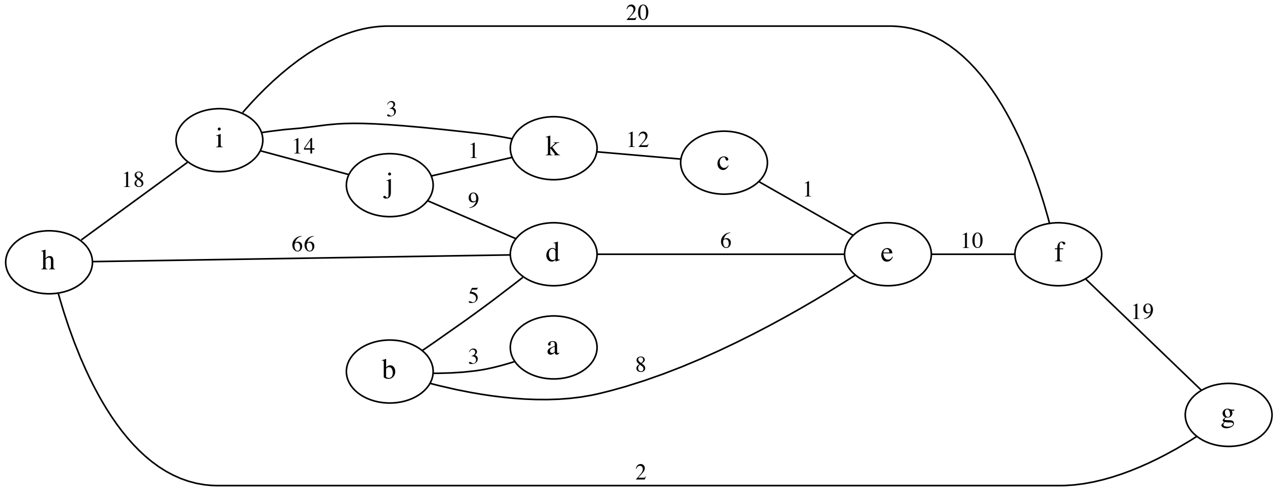


a) Bestimmen Sie für G einen minimalen Spannbaum mit Hilfe des Prim-Algorithmus.

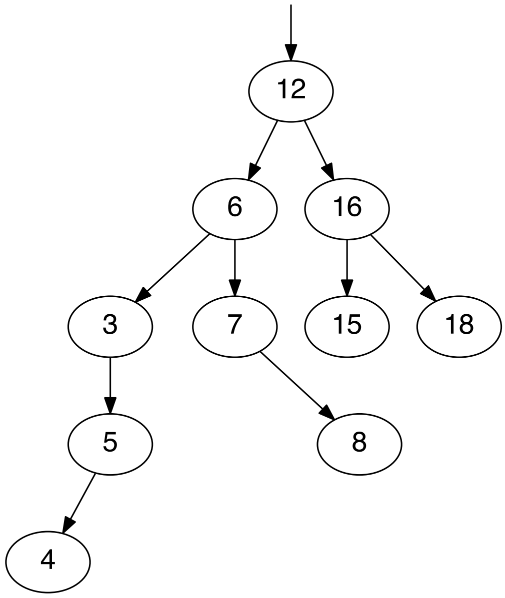
b) Wie viele Kanten enthält der MSB (minimale Spannbaum) für einen Graphen mit n Knoten? Begründen Sie Ihre Antwort.

c) Verwenden Sie den Algorithmus von Dijkstra, um die minimale Entfernung aller Knoten vom Knoten **h** zu bestimmen. Auf der nächsten Seite finden Sie eine Kopie des Graphen.

Fortsetzung



Aufgabe 7 (3+3+5 Punkte)
Gegeben sei der binäre Suchbaum B.



a) Fügen Sie zu B Knoten für die Schlüssel 9, 14 und 17 hinzu.

b) Entfernen Sie aus B (Achtung: dem ursprünglichen Baum, nicht dem Resultat von Aufgabenteil a) die Knoten 3, 12 und 7.

c) Konstruieren Sie einen fast vollständigen Binärbaum (der nicht unbedingt ein binärer Suchbaum werden wird!), der die Knoten $12, 6, 16, 3, 7, 15, 18, 5, 8, 5, 11$ in dieser Reihenfolge (zeilenweise jeweils von links nach rechts auffüllend) enthält. Wenden Sie dann `heapify()` an, um aus dem Baum einen **Min-Heap** zu machen. Zeichnen Sie den Zwischenstand nach jedem kompletten `bubble-down()` eines Wertes.

Fortsetzung

– Ende der Klausur –