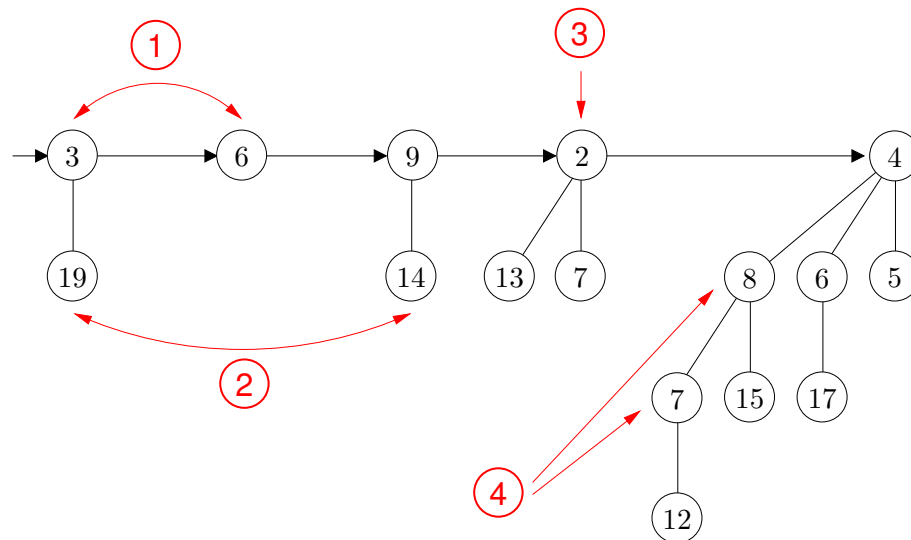


Klausur Algorithmen und Datenstrukturen 2 (Wintersemester 2012/2013)

Lösungshinweise

(Alle Angaben ohne Gewähr¹)

Aufgabe 1.



Der (vermeintliche) Binomial Heap enthält folgende Fehler (siehe Abbildung):

1. Die Binomialbäume sind nicht nach aufsteigendem Knotengrad angeordnet.
2. Der Baum B_1 kommt doppelt vor.
3. Dieser Baum ist kein Binomialbaum.
4. An dieser Stelle wird die Min-Heap Eigenschaft verletzt.

¹Sachdienliche Hinweise zur Fehlerbekämpfung senden Sie bitte an christoph.karg@htw-aalen.de

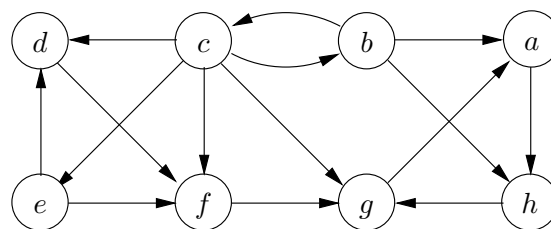
Aufgabe 2. Nach Einfügen der Elemente hat die Hashtabelle folgenden Inhalt:

	87			36	52		100	88		74			29	45	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

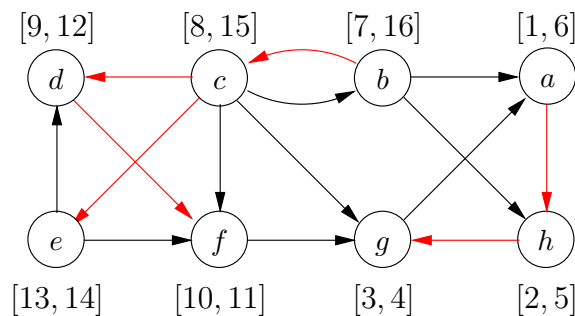
Beim Einfügen werden folgende Slots sondiert:

<i>Schlüssel</i>	<i>Sondierte Slots</i>
36	4
29	13
52	4, 5
88	8
100	4, 5, 7
45	13, 14
74	10
87	7, 8, 10, 13, 1

Aufgabe 3. Zur Berechnung der Zusammenhangskomponenten wird zunächst der transponierte Graph G^T berechnet:



Anschließend wird in G_T eine Tiefensuche durchgeführt, bei der in der Funktion DFS die Knoten nach absteigender Endzeit der ersten Tiefensuche ausgewählt werden:

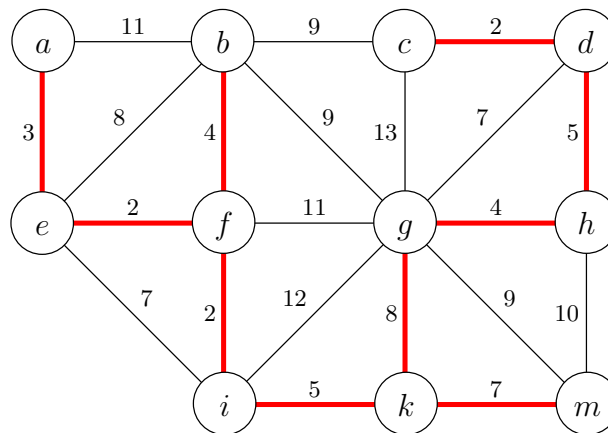


Die starken Zusammenhangskomponenten des Graphen G sind also $\{a, g, h\}$ und $\{b, c, d, e, f\}$.

Aufgabe 4. Der Algorithmus von Kruskal wählt folgende Kanten in genau dieser Reihenfolge aus:

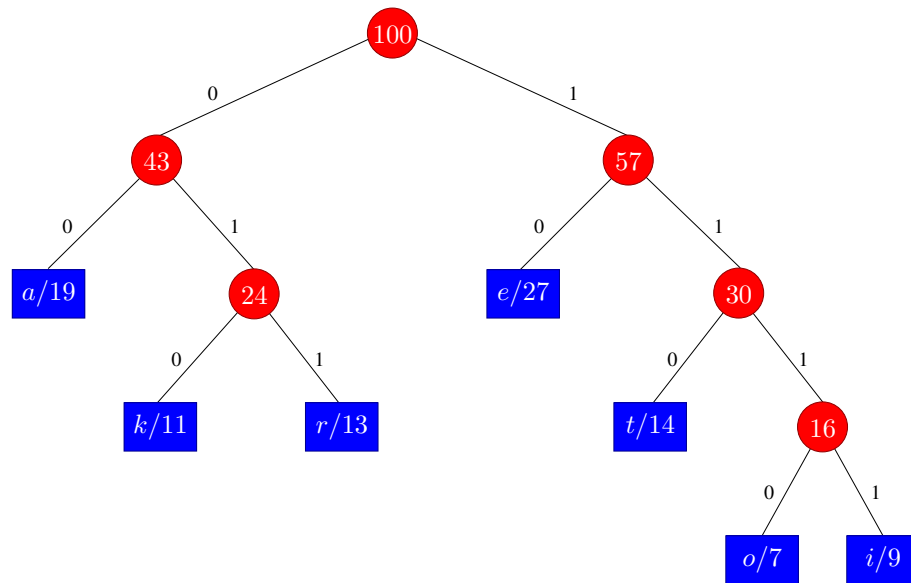
<i>Kante</i>	<i>Knotenmengen</i>
—	$\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}, \{h\}, \{i\}, \{k\}, \{m\}$
(c, d)	$\{a\}, \{b\}, \{c, d\}, \{e\}, \{f\}, \{g\}, \{h\}, \{i\}, \{k\}, \{m\}$
(e, f)	$\{a\}, \{b\}, \{c, d\}, \{e, f\}, \{g\}, \{h\}, \{i\}, \{k\}, \{m\}$
(f, i)	$\{a\}, \{b\}, \{c, d\}, \{e, f, i\}, \{g\}, \{h\}, \{k\}, \{m\}$
(a, e)	$\{a, e, f, i\}, \{b\}, \{c, d\}, \{g\}, \{h\}, \{k\}, \{m\}$
(b, f)	$\{a, b, e, f, i\}, \{c, d\}, \{g\}, \{h\}, \{k\}, \{m\}$
(g, h)	$\{a, b, e, f, i\}, \{c, d\}, \{g, h\}, \{k\}, \{m\}$
(d, h)	$\{a, b, e, f, i\}, \{c, d, g, h\}, \{k\}, \{m\}$
(i, k)	$\{a, b, e, f, i, k\}, \{c, d, g, h\}, \{k\}, \{m\}$
(k, m)	$\{a, b, e, f, i, k, m\}, \{c, d, g, h\}$
(g, k)	$\{a, b, c, d, e, f, g, h, i, k\}$

Das Ergebnis ist folgender minimal aufspannender Baum:



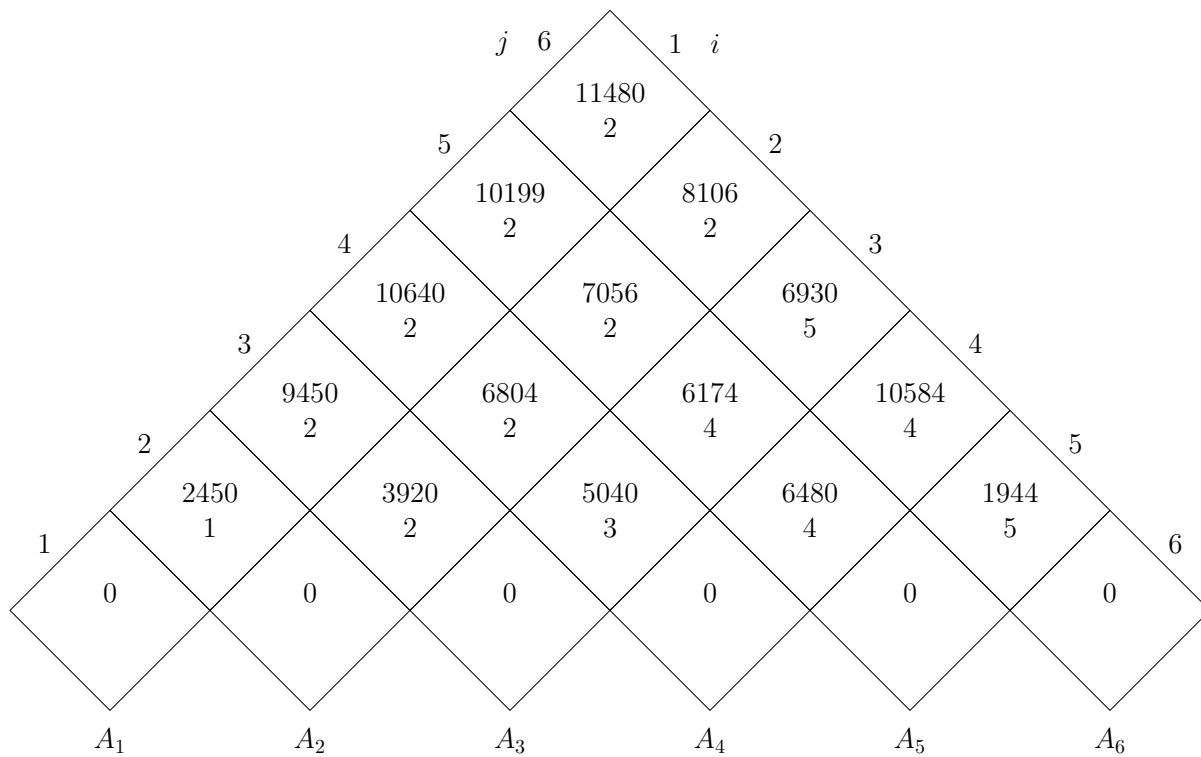
Aufgabe 5.

a) Huffman-Kode:



b) Die Dekodierung des Textes liefert das Wort **korrekt**.

Aufgabe 6. Die vollständige Tabelle ist:



a) $i = 4, j = 5: 0 + 0 + 40 \cdot 18 \cdot 9 = 6480$

b) $i = 2, j = 6:$

$$k = 2 : \quad 0 + 6930 + 14 \cdot 7 \cdot 12 = 8106$$

$$k = 3 : \quad 3920 + 10584 + 14 \cdot 40 \cdot 12 = 21224$$

$$k = 4 : \quad 6804 + 1944 + 14 \cdot 18 \cdot 12 = 11772$$

$$k = 5 : \quad 7056 + 0 + 14 \cdot 9 \cdot 12 = 8568$$

Optimum bei $k = 1$

c) Die optimale Klammerung ist: $(A_1 A_2) (((A_3 A_4) A_5) A_6)$.

Aufgabe 7.

a) Ergebnis des Algorithmus von Dijkstra:

<i>Auswahl</i>	<i>v</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>
<i>Init</i>	$d[v]$	0	∞	∞	∞	∞	∞	∞
	$\pi[v]$	—	—	—	—	—	—	—
<i>a</i>	$d[v]$	0	2	5	∞	∞	∞	∞
	$\pi[v]$	—	<i>a</i>	<i>a</i>	—	—	—	—
<i>b</i>	$d[v]$	0	2	3	9	∞	∞	12
	$\pi[v]$	—	<i>a</i>	<i>b</i>	<i>b</i>	—	—	<i>b</i>
<i>c</i>	$d[v]$	0	2	3	4	9	6	12
	$\pi[v]$	—	<i>a</i>	<i>b</i>	<i>c</i>	<i>c</i>	<i>c</i>	<i>b</i>
<i>d</i>	$d[v]$	0	2	3	4	6	6	12
	$\pi[v]$	—	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>c</i>	<i>b</i>
<i>e</i>	$d[v]$	0	2	3	4	6	6	8
	$\pi[v]$	—	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>c</i>	<i>e</i>
<i>f</i>	$d[v]$	0	2	3	4	6	6	8
	$\pi[v]$	—	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>c</i>	<i>e</i>
<i>g</i>	$d[v]$	0	2	3	4	6	6	8
	$\pi[v]$	—	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>c</i>	<i>e</i>

b) Der kürzeste Pfad von *a* nach *g* ist:

$$a \xrightarrow{2} b \xrightarrow{1} c \xrightarrow{d} c \xrightarrow{2} e \xrightarrow{2} g.$$

Die Länge des Pfades ist 8.

Aufgabe 8. Die Wortlänge der Schlüssel ist $w = 16$. Deshalb muss zunächst $A = \frac{\sqrt{3}}{2}$ durch eine 16-Bit Zahl s approximiert werden:

$$s = \lceil \frac{\sqrt{3}}{2} \cdot 2^{16} \rceil = 56756$$

Zur Berechnung des Hashwerts des Schlüssels $k = 42$ berechnet man

$$\begin{aligned} r &\equiv 56756 \cdot 42 && (\text{mod } 65536) \\ &\equiv 2383752 && (\text{mod } 65536) \\ &= 24456 && (\text{mod } 65536) \end{aligned}$$

Die Größe der Hashtabelle ist $m = 2^{12}$. Deshalb werden die 12 höchstwertigsten Bits von 24456 zur Berechnung des Slots verwendet:

$$24456 = (\underbrace{010111111000}_{=1528} 1000)_2$$

Also ist $h(42) = 1528$.