Prof. Dr. Christoph Karg
Studiengang Informatik
Hochschule Aalen

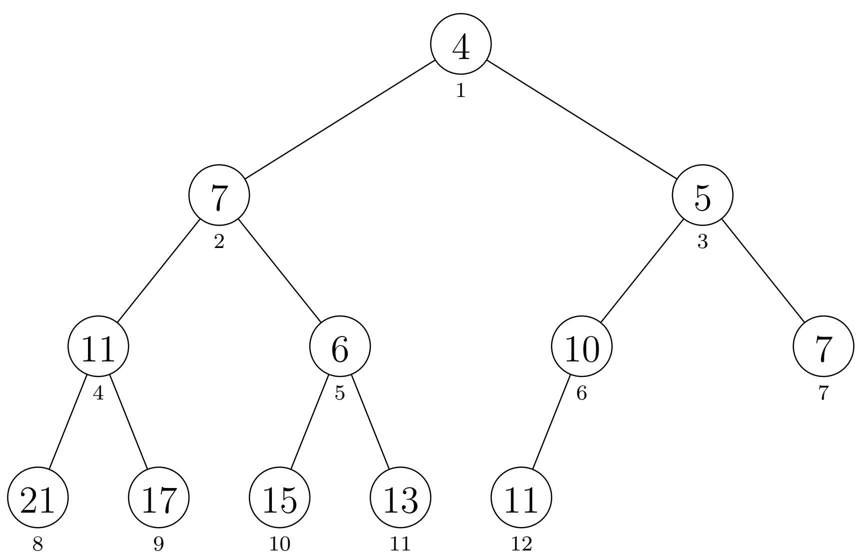
Klausur Algorithmen und Datenstrukturen 2 (Sommersemester 2011)

Lösungshinweise

(Alle Angaben ohne Gewähr[1])

Aufgabe 1.

a) Darstellung des Heaps als Baum:



b) Der Heap ist kein Min-Heap, da $A[2] = 7 > 6 = A[6]$. Dies widerspricht der Anforderung, dass, dass der Schlüssel eines Kindknotens kleiner oder gleich dem Schlüssel des zugehörigen Elternknotens sein muss.

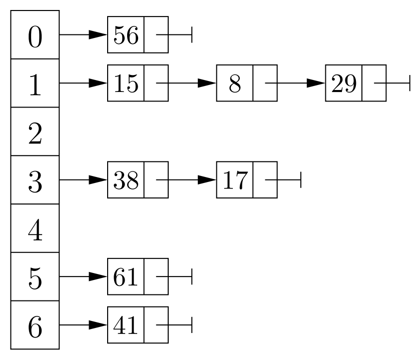
[1]Sachdienliche Hinweise zur Fehlerbekämpfung senden Sie bitte an `christoph.karg@htw-aalen.de`

Aufgabe 2.

a) Die Elemente werden in die folgenden Slots gehasht:

x	56	17	29	38	41	8	15	61
$h(x)$	0	3	1	3	6	1	1	5

Dies führt zu folgender Hash-Tabelle:



b) Multiplikationsmethode ($m = 2048$, $A = \frac{\sqrt{5}-1}{2}$):

$$\begin{aligned} h(753) &= \lfloor (753 \cdot A - \lfloor 753 \cdot A \rfloor) \cdot m \rfloor \\ &= \lfloor (465.3795935 - 465) \cdot 2048 \rfloor \\ &= \lfloor 0.3795935 \cdot 2048 \rfloor \\ &= 777 \end{aligned}$$

c) Bei Double Hashing ist die Hashfunktion definiert als:

$$h(k, i) = (h_1(k) + i \cdot h_2(k)) \bmod 11$$

Für $h_1(k) = k \bmod 11$ und $h_2(k) = 1 + (k \bmod 10)$ sowie den Schlüssel (k=53) ist

$$h_2(53) = 1 + (53 \bmod 10) = 4$$

und

$$h(53, i) = (53 + 4 \cdot i) \bmod 11.$$

Dies führt zu folgender Sondierungssequenz:

i	0	1	2	3	4	5	6	7	8	9	10
$h(53, i)$	9	2	6	10	3	7	0	4	8	1	5

Aufgabe 3.

a) Berechnung der kürzesten Pfade mit dem Dijkstra-Algorithmus:

Auswahl	v	a	b	c	d	e	f	g	h
—	$d[v]$	0	∞	∞	∞	∞	∞	∞	∞
	$\pi[v]$	–	–	–	–	–	–	–	–
a	$d[v]$	0	1	∞	∞	6	15	12	∞
	$\pi[v]$	–	a	–	–	a	a	a	–
b	$d[v]$	0	1	2	∞	6	15	12	∞
	$\pi[v]$	–	a	b	–	a	a	a	–
c	$d[v]$	0	1	2	3	6	15	12	∞
	$\pi[v]$	–	a	b	c	a	a	a	–
d	$d[v]$	0	1	2	3	4	15	12	∞
	$\pi[v]$	–	a	b	c	d	a	a	–
e	$d[v]$	0	1	2	3	4	11	9	∞
	$\pi[v]$	–	a	b	c	d	e	e	–
g	$d[v]$	0	1	2	3	4	11	9	10
	$\pi[v]$	–	a	b	c	d	e	e	h
h	$d[v]$	0	1	2	3	4	11	9	10
	$\pi[v]$	–	a	b	c	d	e	e	h
f	$d[v]$	0	1	2	3	4	11	9	10
	$\pi[v]$	–	a	b	c	d	e	e	h

b) Ein kürzester Pfad von a nach h ist

$$a \xrightarrow{1} b \xrightarrow{1} c \xrightarrow{1} d \xrightarrow{1} e \xrightarrow{5} g \xrightarrow{1} h.$$

Er hat eine Länge von 10 Einheiten.

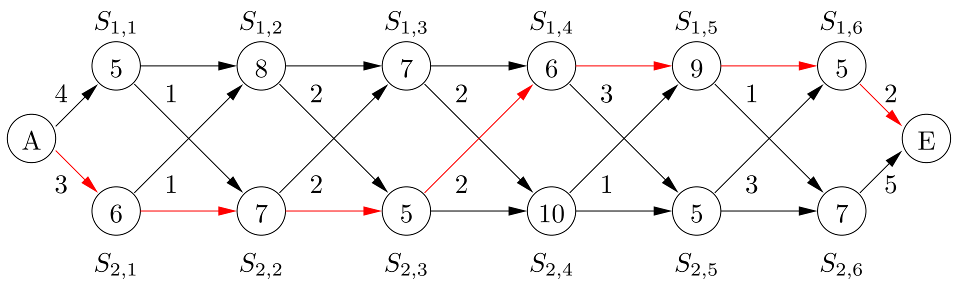
Aufgabe 4.

a) Der Algorithmus zur Berechnung einer optimalen Produktionsroute liefert folgendes Ergebnis:

j	1	2	3	4	5	6	E
$f_1[j]$	9	17	24	29	38	43	45
$f_2[j]$	9	16	21	31	36	43	48

j	2	3	4	5	6
$\ell_1[j]$	1	1	2	1	1
$\ell_2[j]$	2	2	2	2	2

b) Der optimale Produktionspfad ist:

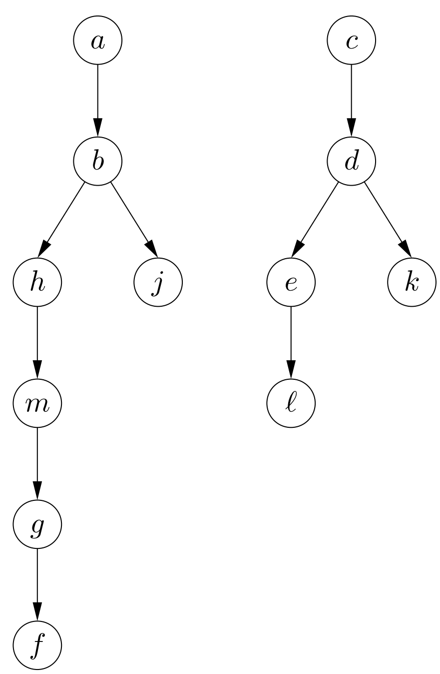


Aufgabe 5.

a) Die Tiefensuche durchläuft die Knoten in der folgenden Reihenfolge:

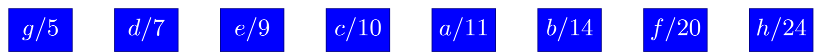
v	$d[v]$	$f[v]$	$\pi[v]$
a	1	14	–
b	2	13	a
h	3	10	b
m	4	9	h
g	5	8	m
f	6	7	g
j	11	12	b
c	15	24	–
d	16	23	c
e	17	20	d
ℓ	18	19	e
k	21	22	d

b) Der Depth-First Wald besteht aus folgenden zwei Bäumen:

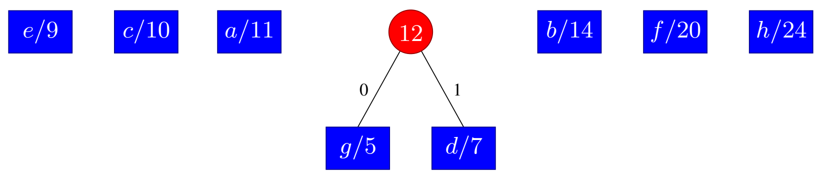


Aufgabe 6. Schrittweise Ausführung des Huffman-Algorithmus:

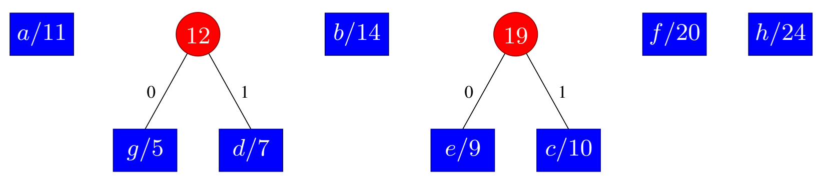
Schritt 1:



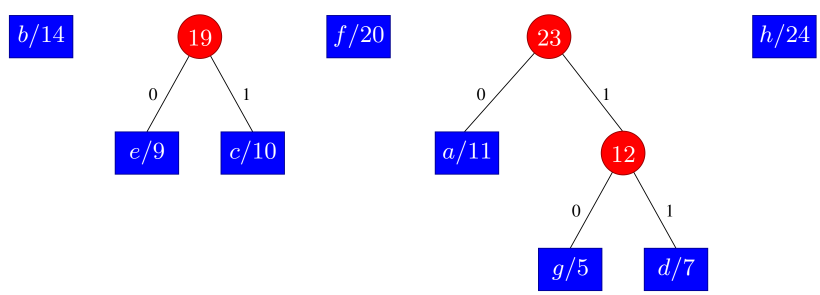
Schritt 2:



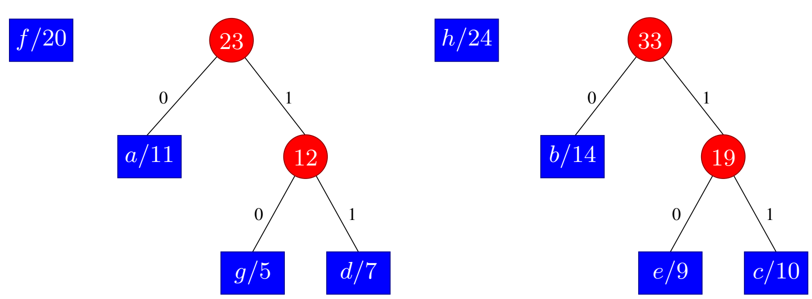
Schritt 3:



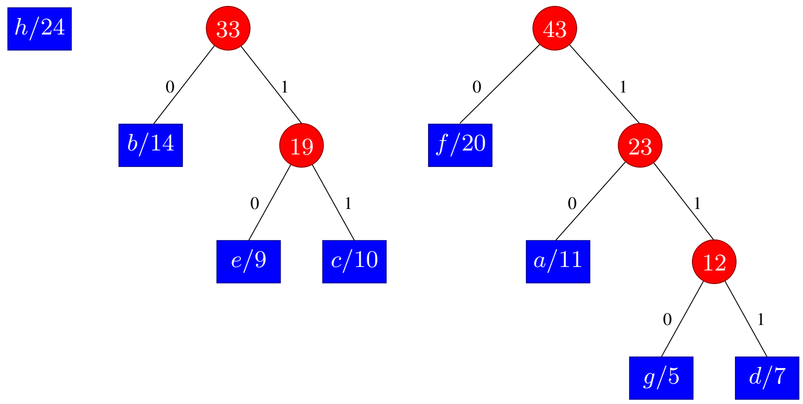
Schritt 4:



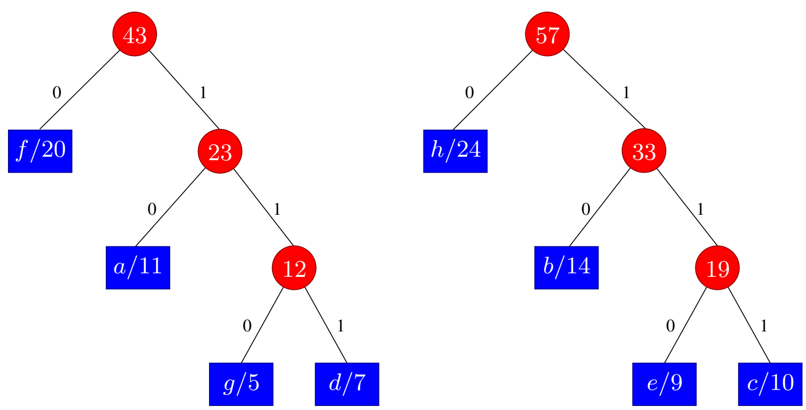
Schritt 5:



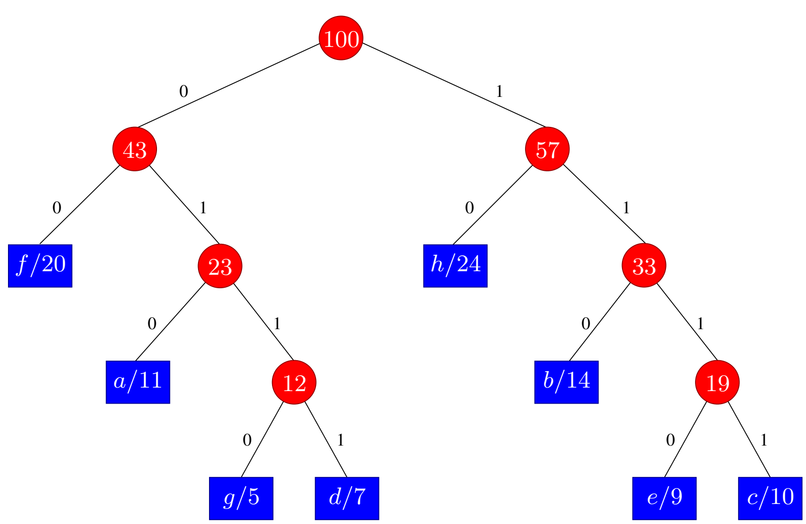
Schritt 6:



Schritt 7:



Schritt 8:

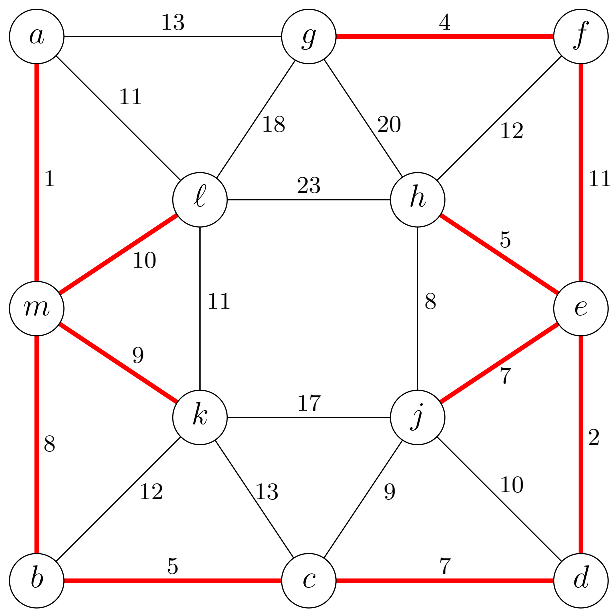


Aufgabe 7.

a) Die Knoten werden in folgender Reihenfolge ausgewählt:

v	a	m	b	c	d	e	h	j	k	ℓ	f	g
$\pi[v]$	$-$	a	m	b	c	d	e	e	m	m	e	f

b) Der minimal aufspannende Baum ist:



Der Baum hat ein Gewicht von 69.

Aufgabe 8. Der Belegungsfaktor einer Hash-Tabelle ist $\alpha = \frac{n}{m}$, wobei n die Anzahl der Elemente in der Tabelle und m die Größe der Tabelle ist. Die mittlere Anzahl der zu untersuchenden Elemente bei einer erfolgreichen Suche ist:

$$1 + \frac{\alpha}{2} - \frac{\alpha}{2n}$$

Für $\alpha = 0.3$ ist die mittlere Anzahl höchstens

$$1 + \frac{0.3}{2} - \frac{0.3}{2 \cdot 307} = 1.149511$$

Für $\alpha = 0.8$ ergibt sich ein Mittelwert von 1.3995116.