Prof. Dr. Christoph Karg
Studiengang Informatik
Hochschule Aalen

Klausur Algorithmen und Datenstrukturen 3 (WS 2009/2010)

Lösungshinweise

(Alle Angaben ohne Gewähr[1])

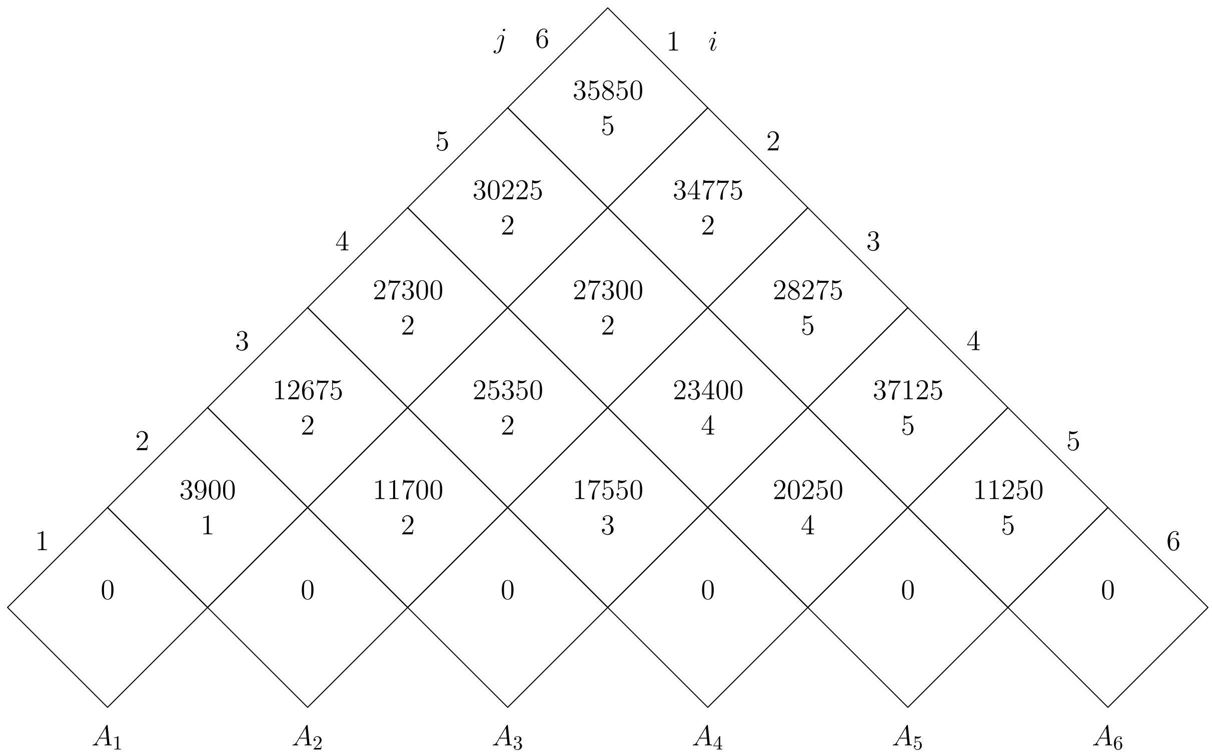
Aufgabe 1. Sortieren der Belegungszeiten nach aufsteigender Endzeit und Auswahl gemäß Algorithmus:

j	s_j	f_j	*Auswahl*
10	1	5	$\star$
2	3	6	
1	6	9	$\star$
5	4	10	
9	7	11	
8	8	12	
12	1	13	
6	9	15	$\star$
7	13	18	
4	15	20	$\star$
11	8	24	
3	22	25	$\star$

Lösung: $\{1, 3, 4, 6, 10\}$

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

Aufgabe 2. Die vollständige Tabelle ist:



a) $i = 3, j = 3$: $0 + 0 + 13 \cdot 45 \cdot 30 = 17550$

b) $i = 1, j = 5$:

$$\begin{array}{lrcl}
k = 2: & 0 + 28275 + 20 \cdot 13 \cdot 25 & = & 34775 \\
k = 3: & 11700 + 37125 + 20 \cdot 45 \cdot 25 & = & 71325 \\
k = 4: & 25350 + 11250 + 20 \cdot 30 \cdot 25 & = & 51600 \\
k = 5: & 27300 + 0 + 20 \cdot 15 \cdot 25 & = & 34800
\end{array}$$

Optimum bei $k = 2$

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

Aufgabe 3.

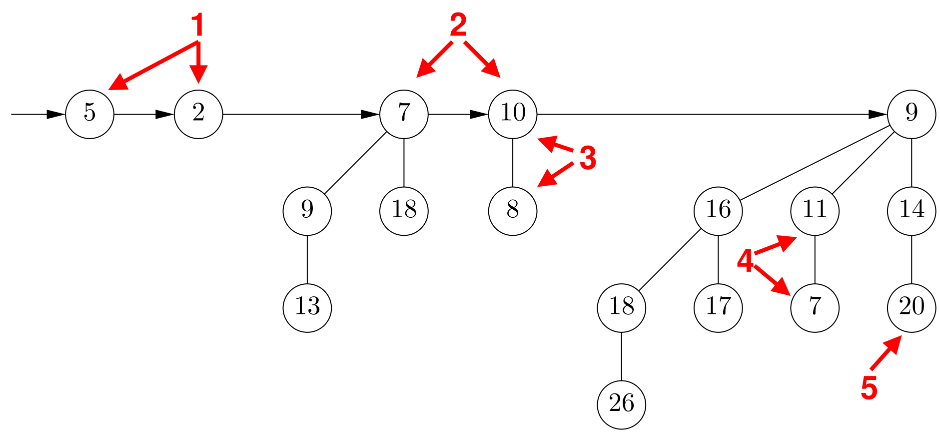
a) Die Tiefensuche durchläuft die Knoten in der folgenden Reihenfolge:

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

b) Kantentypen:

Kante	*Typ*
(a,h)	Tree Edge
(b,c)	Tree Edge
(b,e)	Forward Edge
(c,e)	Tree Edge
(d,c)	Back Edge
(e,d)	Tree Edge
(e,f)	Tree Edge
(e,j)	Forward Edge
(f,a)	Cross Edge
(f,e)	Back Edge
(f,g)	Tree Edge
(f,h)	Cross Edge
(f,j)	Tree Edge
(f,k)	Forward Edge
(g,a)	Cross Edge
(g,b)	Back Edge
(g,f)	Back Edge
(j,k)	Tree Edge
(k,e)	Back Edge

Aufgabe 4.



- (1): Jeder Binomialbaum darf höchstens einmal im Binomial Heap enthalten sein.
- (2): Die Liste der Binomialbäume muss nach aufsteigendem Knotengrad sortiert sein.
- (3)+(4): Die Min Heap Eigenschaft wird verletzt.
- (5): Dieser Knoten verletzt den Aufbau eines Binomialbaums.

Aufgabe 5.

a) Multiplikationsmethode ($m = 1024$): Parameter $A = \frac{\sqrt{5}+1}{2}$:

Nicht geeignet, da $A = 1.6180\ldots > 1$

b) Open Addressing mit Quadratic Probing ($m = 16$): $h(k, i) = (3i^2 + 4i + k) \bmod 16$:

Nicht geeignet, da die Hashfunktion keine korrekte Sondierungssequenz liefert. Betrachte:

$$\begin{aligned} h(k,0) &= k \bmod 16 \\ h(k,4) &= (3 \cdot 4^2 + 4 \cdot 4 + k) \bmod 16 \\ &= (4 \cdot 16 + k) \bmod 16 \\ &= k \bmod 16 \\ &= h(k,0) \end{aligned}$$

Aufgabe 6.

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	∞	∞	4	∞	∞	∞
	$\pi[v]$	–	a	–	–	a	–	–	–
b	$d[v]$	0	1	3	∞	4	2	8	∞
	$\pi[v]$	–	a	b	–	a	b	b	–
f	$d[v]$	0	1	3	∞	3	2	8	∞
	$\pi[v]$	–	a	b	–	f	b	b	–
c	$d[v]$	0	1	3	5	3	2	5	12
	$\pi[v]$	–	a	b	c	f	b	c	c
e	$d[v]$	0	1	3	5	3	2	5	12
	$\pi[v]$	–	a	b	c	f	b	c	c
d	$d[v]$	0	1	3	5	3	2	5	8
	$\pi[v]$	–	a	b	c	f	b	c	d
g	$d[v]$	0	1	3	5	3	2	5	8
	$\pi[v]$	–	a	b	c	f	b	c	d
h	$d[v]$	0	1	3	5	3	2	5	8
	$\pi[v]$	–	a	b	c	f	b	c	d

b) Ein kürzester Pfad von a nach h ist

$$a \xrightarrow{1} b \xrightarrow{2} c \xrightarrow{2} d \xrightarrow{3} h.$$

Er hat eine Länge von 8 Einheiten.

Aufgabe 7.

a) $m = 2^{10} = 1024,\ A = \frac{\pi^2}{11},\ k = 12345$:

$$\begin{aligned} h(k) &= \lfloor (k \cdot A - \lfloor k \cdot A \rfloor) \cdot m \rfloor \\ h(12345) &= \left\lfloor \left(12345 \cdot \frac{\pi^2}{11} - \left\lfloor 12345 \cdot \frac{\pi^2}{11} \right\rfloor \right) \cdot 4096 \right\rfloor \\ &= 1588 \end{aligned}$$

b) Berechnung der Sondierungssequenz unter Verwendung des Algorithmus aus der Vorlesung:

j	i
0	4
1	5
2	7
3	10
4	14
5	3
6	9
7	0
8	8
9	1
10	11
11	6
12	2
13	15
14	13
15	12

Die Sondierungsequenz lautet:

$$\langle 4, 5, 7, 10, 14, 3, 9, 0, 8, 1, 11, 6, 2, 15, 13, 12 \rangle$$