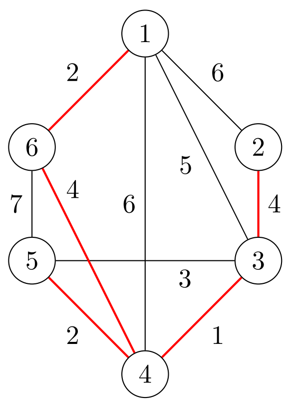
Einführung in die Logistik, Übung 3

Henry Haustein

Aufgabe 6

(a) Graph



(b) Algorithmus von Prim

- $W = \{3\}$, $V = \{1, 2, 4, 5, 6\}$, $E^* = \{[3, 4]\}$, $W = \{3, 4\}$, $V = \{1, 2, 5, 6\}$
- wähle $[4, 5]$: $E^* = \{[3, 4], [4, 5]\}$, $W = \{3, 4, 5\}$, $V = \{1, 2, 6\}$
- wähle $[4, 6]$: $E^* = \{[3, 4], [4, 5], [4, 6]\}$, $W = \{3, 4, 5, 6\}$, $V = \{1, 2\}$
- wähle $[6, 1]$: $E^* = \{[3, 4], [4, 5], [4, 6], [6, 1]\}$, $W = \{1, 3, 4, 5, 6\}$, $V = \{2\}$
- wähle $[3, 2]$: $E^* = \{[3, 4], [4, 5], [4, 6], [6, 1], [3, 2]\}$, $W = \{1, 2, 3, 4, 5, 6\}$, $V = \emptyset \Rightarrow$ Ende

Aufgabe 7

Es gibt eine sehr schöne Schreibweise für den Ablauf des Algorithmus aus der Vorlesung *Einführung in die Informatik*. Dabei ist die Menge S die Menge der bisher erkundeten Punkte, zu diesen Punkten kennt man den kürzesten Weg und $D(i)$ ist die (kumulierte) Distanz zum Knoten i. Die <u>unterstrichene</u> Distanz ist die kürzeste Distanz zu einem noch nicht erkundeten Punkt

Iteration	S	$D(1)$	$D(2)$	$D(3)$	$D(5)$	$D(6)$
0	$\{4\}$	∞	4	6	$\underline{2}$	∞
1	$\{4, 5\}$	∞	$\underline{4}$	6	2	∞
2	$\{2, 4, 5\}$	$\underline{6}$	4	6	2	∞
3	$\{1, 2, 4, 5\}$	6	4	$\underline{6}$	2	7
4	$\{1, 2, 3, 4, 5\}$	6	4	6	2	$\underline{7}$
5	$\{1, 2, 3, 4, 5, 6\}$	6	4	6	2	7

Aufgabe 8

(a) Die Bewertungsmatrix lautet

$$C(\vec{G}) = \begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & 0 & 340 & 530 & \infty & \infty & \infty \\ 2 & \infty & 0 & 280 & 750 & \infty & \infty \\ 3 & \infty & \infty & 0 & 570 & 480 & \infty \\ 4 & \infty & \infty & \infty & 0 & 460 & 810 \\ 5 & \infty & \infty & \infty & \infty & 0 & 840 \\ 6 & \infty & \infty & \infty & \infty & \infty & 0 \end{array}$$

(b) Anwendung des Verfahrens von Bellmann liefert

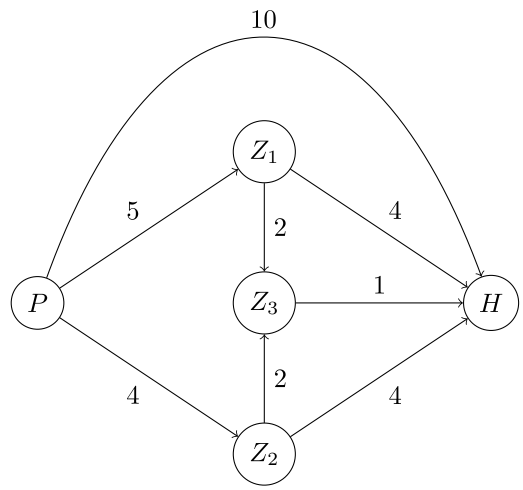
j	d_{1j}	$V(j)$
1	0	1
2	340	1
3	530	1
4	1090	2
5	1100	3
6	1900	4

(c) Arbeitet man die Tabelle von unten nach oben durch, so ergibt sich $6 \leftarrow 4 \leftarrow 2 \leftarrow 1$, also ist der kürzeste Weg: $1 \rightarrow 2 \rightarrow 4 \rightarrow 6$.

(d) Baumalgorithmen ermitteln den kürzesten Weg von einem Startknoten zu allen anderen Knoten, z.B. Dijkstra-Algorithmus
Matrixalgorithmen ermitteln den kürzesten Weg von jedem Knoten zu jedem Knoten, z.B. Tripelalgorithmus

Aufgabe 9

(a) Graph



Die Vorgängermatrix lautet

W^0	P	Z_1	Z_2	Z_3	H
P	P	P	P	∞	P
Z_1	∞	Z_1	∞	Z_1	Z_1
Z_2	∞	∞	Z_2	Z_2	Z_2
Z_3	∞	∞	∞	Z_3	Z_3
H	∞	∞	∞	∞	H

(b) Die Entfernungsmatrix und die Wegematrix lauten

D	P	Z_1	Z_2	Z_3	H
P	0	5	4	6	7
Z_1	∞	0	∞	2	4
Z_2	∞	∞	0	2	4
Z_3	∞	∞	∞	0	1
H	∞	∞	∞	∞	0

W	P	Z_1	Z_2	Z_3	H
P	P	P	P	Z_2	Z_2Z_3
Z_1	∞	Z_1	∞	Z_1	Z_1
Z_2	∞	∞	Z_2	Z_2	Z_2
Z_3	∞	∞	∞	Z_3	Z_3
H	∞	∞	∞	∞	H

Die Entfernungsmatrix gibt die Entfernung zwischen 2 Punkten an und die Wegematrix den Weg.

(c) Mit der selben Notation wie in Aufgabe 7 folgt

Iteration	S	$D(Z_1)$	$D(Z_2)$	$D(Z_3)$	$D(H)$
0	$\{P\}$	5	$\underline{4}$	∞	10
1	$\{P, Z_2\}$	$\underline{5}$	4	6	8
2	$\{P, Z_1, Z_2\}$	5	4	$\underline{6}$	8
3	$\{P, Z_1, Z_2, Z_3\}$	5	4	6	$\underline{7}$
4	$\{P, Z_1, Z_2, Z_3, H\}$	5	4	6	7