Fachhochschule Bonn-Rhein-Sieg

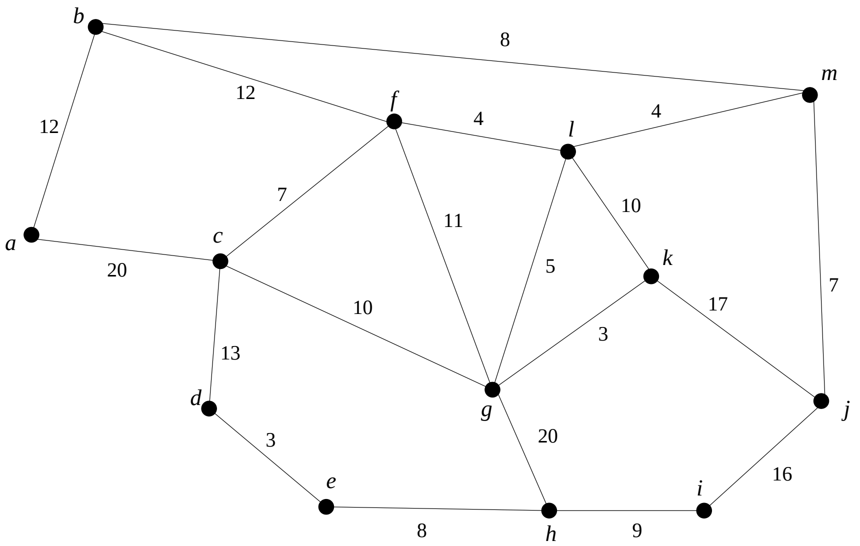
Fachbereich Informatik
Prof. Dr. Peter Becker

Vorlesung Graphentheorie
Wintersemester 2007/08
23. November 2007

Lösungen zu Aufgabenblatt 4

Aufgabe 1 (Kürzeste Wege)

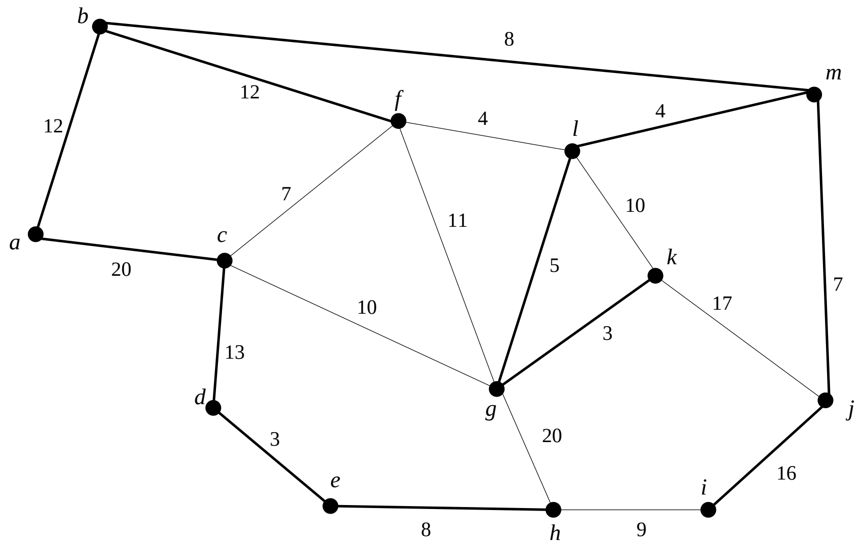
Berechnen Sie für das unten angegebene Netzwerk die Abstände und die kürzesten Wege aller Knoten vom Knoten a.



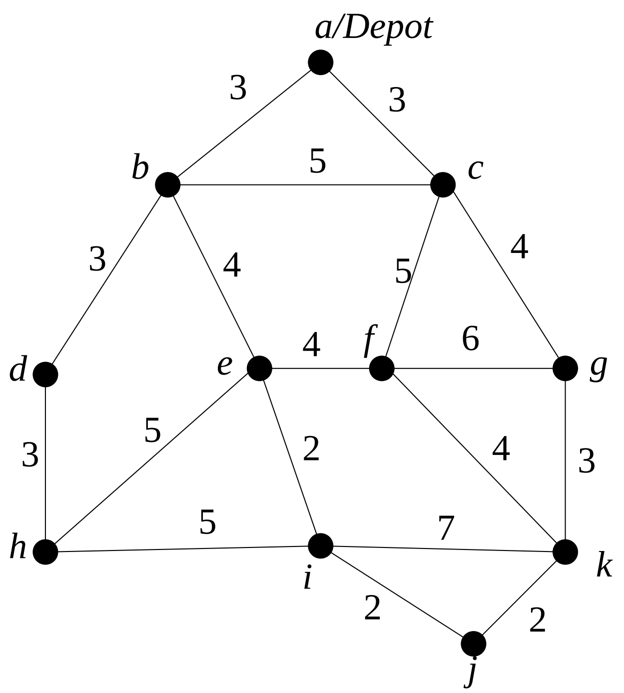
Lösung:

Iteration	a	b	c	d	e	f	g	h	i	j	k	l	m	u	d(u)	p(u)
1	0	–	–	–	–	–	–	–	–	–	–	–	–	a	0	–
2		12	20	–	–	–	–	–	–	–	–	–	–	b	12	a
3			20	–	–	24	–	–	–	–	–	–	20	c	20	a
4				33	–	24	30	–	–	–	–	–	20	m	20	b
5				33	–	24	30	–	–	27	–	24		f	24	b
6				33	–		30	–	–	27	–	24		l	24	m
7				33	–		29	–	–	27	34			j	27	m
8				33	–		29	–	43		34			g	29	l
9				33	–			49	43		32			k	32	g
10				33	–			49	43					d	33	c
11					36			49	43					e	36	d
12								44	43					i	43	j
13								44						h	44	e

Gerüst mit den kürzesten Wegen:



Aufgabe 2 (Müllabfuhr)



Der angegebene Graph modelliert ein Straßennetz, die Kantenbewertungen geben die Straßenlängen an.

Ermitteln Sie eine möglichst kurze Route für die Müllabfuhr in diesem Straßennetz. Anforderungen:

- Das Müllauto muß im Depot starten und wieder zu diesem zurückkehren.
- Jede Straße muß mindestens einmal durchfahren werden.

Machen Sie die Herleitung Ihrer Route deutlich.

Lösung:

Der theoretische Idealfall würde vorliegen, wenn das Straßennetz einen eulerschen Graphen bildet. Dann wäre ein Eulerkreis die optimale Lösung.

Der gegebene Graph ist aber nicht eulersch, denn er enthält zwei Knoten (g und h) mit ungeradem Grad. Zu einer optimalen Lösung kommen wir, indem wir zwischen diesen Knoten eine zusätzliche Kante einfügen. Diese Kante muß dabei dem kürzesten Weg von g nach h entsprechen.

Mit dem Algorithmus von Dijkstra können wir den kürzesten Weg von g nach h berechnen. Dies ist (g, k, j, i, h) mit der Länge 12.

Wir berechnen nun für den Graphen, der zusätzlich die Kante $\{g, h\}$ enthält, einen Eulerkreis (z.B. mit dem Algorithmus von Hierholzer). Wir erhalten:

$$(a, b, d, h, i, j, k, g, c, b, e, f, g, h, e, i, k, f, c, a)$$

Für die Kante (g, h) setzen wir den zuvor ermittelten kürzesten Weg ein und erhalten somit als optimale Route:

$$(a, b, d, h, i, j, k, g, c, b, e, f, g, k, j, i, h, e, i, k, f, c, a)$$

Aufgabe 3 (Textfaktorisierung)

Die Nachricht `babbbaabba` soll mit dem folgenden Codebuch codiert werden:

Text	Code	Länge
a	00	2
b	010	3
ba	0110	4
bb	0111	4
abb	1	1

Beispiel für eine mögliche Codierung der Nachricht:

ba	bb	ba	a	bb	a
0110	0111	0110	00	0111	00

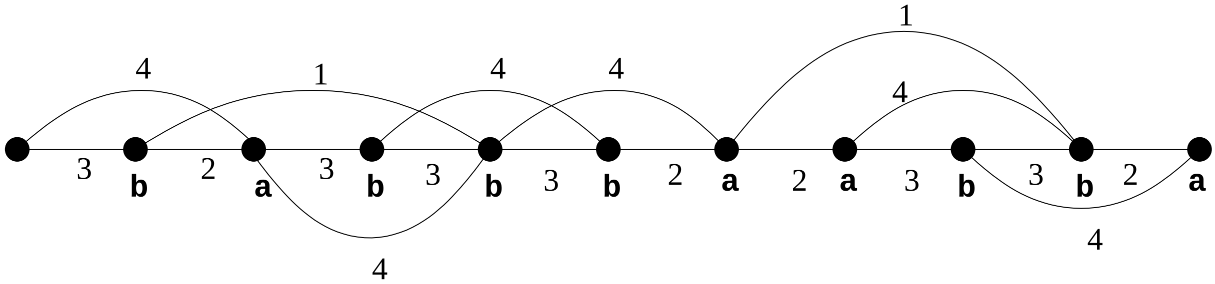
Die Gesamtlänge dieser Codierung ist 20.

Finden Sie eine Codierung mit minimaler Gesamtlänge. Erläutern Sie hierzu kurz, wie Sie dieses Problem mit Hilfe der Graphentheorie lösen können und berechnen Sie eine Lösung.

Lösung:

Sei $text$ = `babbbaabba` $= c_1 \ldots c_{10}$. Wir definieren einen gerichteten azyklischen Graphen $G = (V, A)$ mit $V = \{0, \ldots 10\}$ und $A = \{(i, j) | c_{i+1} \ldots c_j$ ist im Codebuch$\}$.

Die Länge $w(e)$ einer gerichteten Kante $e \in A$ ergibt sich durch die Länge der zugehörigen Codierung im Codebuch.



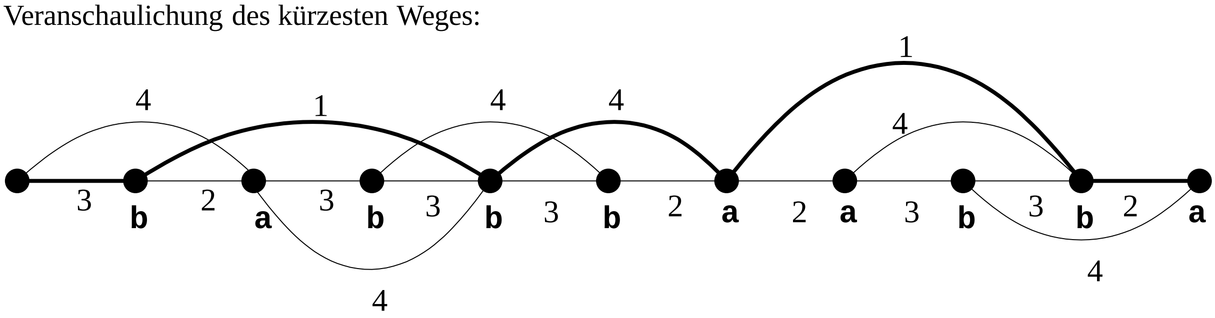
Jeder Weg von 0 nach 10 legt eine Codierung fest. Somit erhält man eine Codierung mit minimaler Gesamtlänge durch einen kürzesten Weg von 0 nach 10. Dieser kann mit dem Algorithmus von Dijkstra oder durch die Rekursionformel gemäß Satz 3.6 bestimmt werden.

Für die Rekursionsformel erhalten wir:

v	$d(v)$	$best(v)$	Text
0	0	-	-
1	3	0	b
2	4	0	ba
3	7	2	b
4	4	1	abb
5	7	4	b
6	8	4	ba
7	10	6	a
8	13	7	b
9	9	6	abb
10	11	9	a

Damit ist $\begin{array}{ccccc} \texttt{b} & \texttt{abb} & \texttt{ba} & \texttt{abb} & \texttt{a} \\ 010 & 1 & 0110 & 1 & 00 \end{array}$ eine Codierung mit minimaler Gesamtlänge 11.

Veranschaulichung des kürzesten Weges:



Aufgabe 4 (Einkaufsplanung)

Wir betrachten einen Händler, der über n Perioden ein Produkt zu einem variablen Preis einkauft und zu einem festen Preis verkauft. Der Händler hat eine beschränkte Lagerkapazität, wobei das Lagern Kosten pro Mengeneinheit verursacht. Es soll ein möglichst günstiger Einkaufsplan erstellt werden.

- p_i sei der Einkaufspreis für das Produkt in Periode $i = 0, \ldots, n$
- d_i sei die Nachfrage in Periode $i = 1, \ldots, n$
- x_i gebe die Mengeneinheiten an, die in Periode $i = 0, \ldots, n$ zum Preis p_i eingekauft werden.
- b_i sei der Lagerbestand zu Beginn von Periode $i = 0, \ldots, n$. Es gelte $b_0 = 0$. und

  $$b_{i+1} = b_i - d_i + x_i \text{ für } i = 1, \ldots, n$$

  Der Bestand darf nie negativ werden, d.h. es muß $b(i) \geq 0$ für alle $i = 1, \ldots, n+1$ gelten. Weiterhin ist die Lagerkapazität beschränkt auf L Mengeneinheiten, d.h. $b(i) \leq L$ für $i = 1, \ldots, n$.
- Pro Mengeneinheit fallen h Geldeinheiten Lagerhaltungskosten an, d.h. die Lagerhaltungskosten in Periode i sind $h \cdot b(i)$.

- Man bestimme die Größen x_i für $i = 0, \dots, n$, so daß die Gesamtkosten
$$\sum_{i=0}^{n} p_i \cdot x_i + \sum_{i=1}^{n+1} h \cdot b_i$$
minimal sind unter der Nebenbedingung, daß die Bestände nie negativ werden und die Lagerkapazität nicht überschritten wird (siehe oben).

Zeigen Sie, wie Sie dieses Problem in ein Kürzestes-Wege-Problem überführen können und lösen Sie es für die folgenden Daten.

$$h = 1, L = 5$$

i	0	1	2	3	4	5
$d(i)$	–	2	1	4	3	5
$p(i)$	5	6	4	10	9	12

Hinweis: Repräsentieren Sie eine Kombination aus Periode und Lagerbestand als Knoten.

Lösung:

Wie im Hinweis beschrieben stellt eine Kombination aus Periode und Lagerbestand einen Knoten dar. Ein Knoten ist also ein Tupel (i, b), wobei i die Periode bezeichnet und b den Lagerbestand. In Periode 0 beginnen wir mit einem leeren Lagerbestand. Eine optimale Lösung wird stets dafür sorgen, daß nach der letzten Periode das Lager wieder leer ist. Daher:

$$V = \{(0,0)\} \cup \{(i,b) | i \in \{1,\dots,5\}, b \in \{0,\dots,5\}\} \cup \{(6,0)\}$$

Ein Knoten beschreibt somit einen Zustand, nämlich die Zeitperiode in Verbindung mit dem Lagerbestand. Von Periode i nach Periode $i+1$ können wir nun den Lagerbestand ändern. Wenn wir in Periode i genau $d(i)$ Mengeneinheiten einkaufen, dann bleibt der Lagerbestand unverändert, kaufen wir mehr als $d(i)$, dann steigt der Lagerbestand, kaufen wir weniger als $d(i)$, dann sinkt der Lagerbestand. Wir legen also eine Kanten zwischen zwei Knoten (i, b_i) und $(i+1, b_{i+1})$, wenn solch ein Zustandsübergang möglich ist. Das Gewicht dieser Kante ist $p_i \cdot (b_{i+1} - b_i + d_i) + h \cdot b_{i+1}$.

Beispiel: Wenn in Periode 1 der Lagerbestand 2 vorliegt, müssen wir 3 Mengeneinheiten einkaufen um in Periode 2 einen Lagerbestand von 3 zu haben. Da der Preis für eine Mengeneinheit in Periode 1 6 Geldeinheiten beträgt, sind damit Kosten von 18 Geldeinheiten verbunden. Hinzu kommen die Kosten der Lagerung von $h \cdot b_{i+1}$ der folgenden Periode.

Ein kürzester Weg von Knoten $(0,0)$ nach $(6,0)$ beschreibt dann eine optimale Einkaufsplanung. Die Berechnung dieses kürzesten Weges ist eine Aufgabe von Aufgabenblatt 5.

Die folgenden Tabellen stellen die Berechung dar.

Zustand	Kosten	opt. Vorgänger
(1,0)	0	(0,0)
(1,1)	5+1=6	(0,0)
(1,2)	10+2=12	(0,0)
(1,3)	15+3=18	(0,0)
(1,4)	20+4=24	(0,0)
(1,5)	25+5=30	(0,0)

Zustand	Kosten	opt. Vorgänger
(2,0)	0+12+0=12	(1,0)
(2,1)	6+12+1=19	(1,1)
(2,2)	12+12+2=26	(1,2)
(2,3)	18+12+3=33	(1,3)
(2,4)	24+12+4=40	(1,4)
(2,5)	30+12+5=47	(1,5)

Zustand	Kosten	opt. Vorgänger
(3,0)	12+4+0=16	(2,0)
(3,1)	12+8+1=21	(2,0)
(3,2)	12+12+2=26	(2,0)
(3,3)	12+16+3=31	(2,0)
(3,4)	12+20+4=36	(2,0)
(3,5)	12+24+5=41	(2,0)

Zustand	Kosten	opt. Vorgänger
(4,0)	36+0+0=36	(3,4)
(4,1)	41+0+1=42	(3,5)
(4,2)	41+10+2=53	(3,5)
(4,3)	41+20+3=64	(3,5)
(4,4)	41+30+4=75	(3,5)
(4,5)	41+40+5=86	(3,5)

Zustand	Kosten	opt. Vorgänger
(5,0)	42+18+0=60	(4,1)
(5,1)	42+27+1=70	(4,1)
(5,2)	42+36+2=80	(4,1)
(5,3)	42+45+3=90	(4,1)
(5,4)	42+54+4=100	(4,1)
(5,5)	42+63+5=110	(4,1)

Für den Zustand (6,0) ist dann (5,5) der optimale Vorgänger mit Kosten 110+0+0. Somit ist die folgende Einkaufspolitik optimal:

i	0	1	2	3	4	5
x_i	0	2	6	0	7	0

Aufgabe 5 (Scheduling)

Gegeben sei eine Menge J von Jobs sowie Mengen $P(j)$ für $j \in J$, die angeben, welche Jobs beendet sein müssen, bevor j begonnen werden kann.

j	$P(j)$
X	$\emptyset$
Y	$\emptyset$
P	$\{X, Y\}$
Q	$\{P\}$
R	$\{P\}$
S	$\{R, Q\}$
T	$\{R\}$
U	$\{T, S\}$

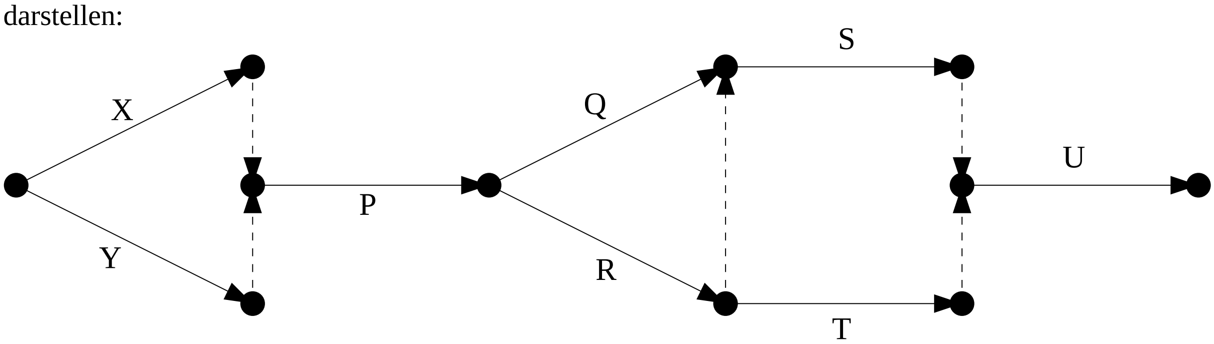
Die Bearbeitung eines Jobs benötigt eine Zeiteinheit. Jeder Job benötigt zur Bearbeitung eine Maschine. Es stehen beliebig viele Maschinen zur Verfügung, so daß Jobs parallel ausgeführt werden können.

Wieviele Zeiteinheiten benötigt man mindestens, um die Jobs korrekt auszuführen? Geben Sie auch an, wieviele Maschinen man für eine möglichst schnelle Ausführung mindestes benötigt.

Was ändert sich an der Berechnung, wenn die Jobs Q, R, T und U statt einer nun zwei Zeiteinheiten benötigen?

Lösung:

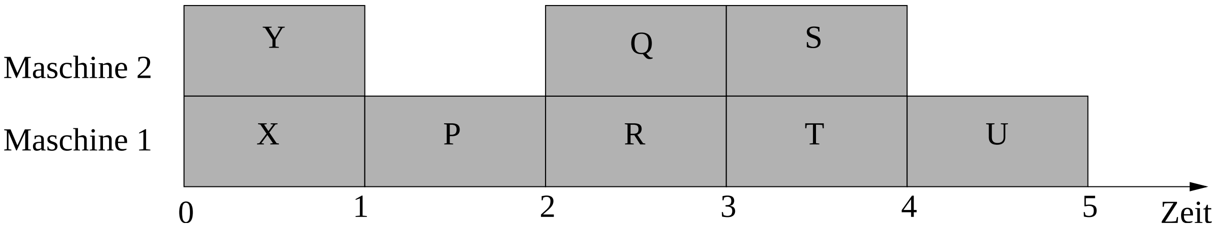
Das Problem entspricht der Projektplanung. Wir können die Abhängigkeiten der Jobs in einem Netzplan darstellen:



Man beachte, daß die gestrichelten Kanten sogenannte Dummy-Vorgänge darstellen, die die Dauer 0 haben. Damit hat der längste Weg (kritischer Pfad) in diesem Netzplan die Länge 5. Wir brauchen also fünf Zeiteinheiten zur Ausführung der Jobs.

Man sieht der Darstellung weiterhin an, daß höchstens zwei Jobs parallel ausgeführt werden können. Demnach brauchen wir zwei Maschinen, um die Jobs innerhalb von fünf Zeiteinheiten ausführen zu können. Zusätzliche Maschinen beschleunigen die Ausführung nicht!

Die folgende Grafik zeigt eine optimale Ausführung als sogenanntes *Gantt-Diagramm*, wie es häufig in der Produktionsplanung verwendet wird. Das Diagramm gibt an, wann welcher Job auf welcher Maschine ausgeführt wird.



Wenn die Jobs Q, R, T und U eine Dauer von zwei Zeiteinheiten haben, hat ein kritischer Pfad die Länge 8 (z.B. (X, P, R, T, U)). Das folgende Gantt-Diagramm veranschaulicht eine optimale Ausführung.