

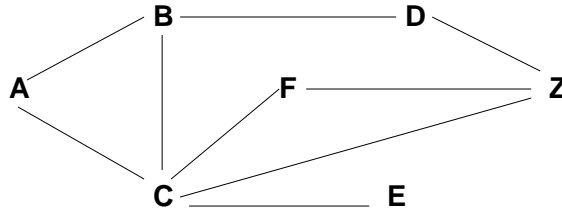
Name:

Matrikelnummer:

Aufgabe 1: Suchalgorithmen

(12 Punkte)

(a) (4 Punkte) Gegeben ist folgendes Streckennetz:



Zeigen Sie durch Handsimulation wie mit dem Tiefensuch-Algorithmus ein (nicht notwendig optimaler) Weg von A nach Z gefunden wird. Für die Reihenfolge, in der Nachfolger-Knoten besucht werden, sei eine **alphabetische Ordnung** vorgegeben ($A < B < C < D < E < F < Z$).

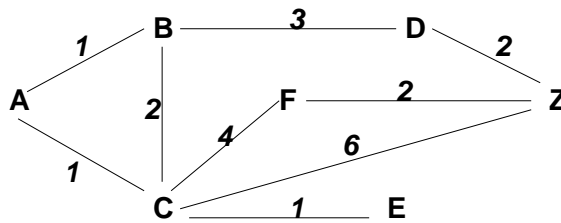
(b) (1 Punkt) Wie kann der Tiefensuch-Algorithmus zum Finden einer Lösung so erweitert werden, daß die optimale Lösung gefunden wird?

Punkte	
--------	--

Name:

Matrikelnummer:

(c) (4 Punkte) Nun sind zusätzlich Entfernungen zwischen den Orten gegeben:



Zeigen Sie durch Handsimulation wie mit dem Branch-and-Bound Algorithmus für bewertete Kanten der Weg von *A* nach *Z* gefunden wird. Die Ordnung der Nachfolger-Knoten sei wieder alphabetisch. (Optional kann, wenn für ein Knotenpaar mehr als ein Weg erzeugt wurde, jeweils nur der am besten bewertete Weg behalten werden.)

Punkte	
--------	--

Name:

Matrikelnummer:

- (d) (1 Punkt) Welche(r) der folgenden Suchalgorithmen für bewertete Kanten findet/finden garantiert die optimale Lösung? Branch-and-Bound, A*, Gradientenmethode (Hill Climbing), best-first-search
- (e) (1 Punkt) Was ist der *worst-case* Aufwand bei der Breitensuche?
- (f) (1 Punkt) Was bewirkt die Verwendung einer unteren Schranke zur Schätzung der Restwegkosten bei A*?

Punkte	
--------	--

Name:

Matrikelnummer:

Aufgabe 2: Greedy-Algorithmen**(13 Punkte)**

(a) (1 Punkt) Wann ist ein Problem greedy-lösbar?

(b) (10 Punkte) In einem Laden sollen Waren mit einer Diebstahlssicherung versehen werden. Da solche Sicherungen teuer sind, sollen nur die besonders gefährdeten Produkte gesichert werden. Als besonders gefährdet werden Produkte angesehen, die (a) besonders teuer sind und (b) geringes Gewicht haben. Die Auswahl soll durch einen *greedy*-Algorithmus erfolgen. Entsprechend wird als *greedy*-Kriterium der Quotient $Preis/Gewicht$ verwendet.

Gegeben seien n Waren und m Diebstahlssicherungen ($m < n$). Die Waren werden durch folgenden Datentyp repräsentiert:

```
TYPE Warentyp = RECORD
    Kennung      : INTEGER;
    Preis        : REAL;
    Gewicht      : REAL;
    Sicherung    : BOOLEAN;
END;
```

```
TYPE Waren = ARRAY[1..n] OF Warentyp; (* n als Konstante vorge-
geben *)
```

Gesucht ist eine (MODULA-2) Prozedur `PROCEDURE sichere (VAR AlleWaren:Waren)`, die aus den n vorhandenen Waren diejenigen m Waren auswählt, die bevorzugt mit einer Diebstahlssicherung versehen werden sollen. Dabei sei `AlleWaren` bereits vorbelegt (Kennung, Preis und Gewicht sind gegeben; Sicherung ist für alle Waren auf `FALSE` gesetzt) und m sei als Konstante gegeben. HINWEIS: Für den *greedy*-Algorithmus ist es **nicht** notwendig, daß die Daten vorsortiert werden!

Punkte	
--------	--

Name:

Matrikelnummer:

Punkte	
--------	--

Name:

Matrikelnummer:

- (c) (2 Punkte) Ein Kind will sein Sparschwein schlachten und das Geld in Bauklötzen anlegen. Natürlich möchte das Kind möglichst viele Bauklötze für sein Geld bekommen. Die Bauklötze gibt es in unterschiedlich großen Abpackungen:

	Preis	Preis pro Stück
120 Stk:	5,00 DM	4,167 Pf
15 Stk:	1,00 DM	6,667 Pf
3 Stk:	0,25 DM	8,333 Pf

- (i) Das Problem ist *greedy*-lösbar. Geben Sie ein *greedy*-Kriterium für das Problem an.

- (ii) Nun werden die Bauklötze in zwei weiteren Abpackungen angeboten: 1 Stück und 40 Stück. Geben Sie mögliche Packungs- und Stückpreise an, so daß das Problem weiter *greedy* lösbar ist.

	Preis	Preis pro Stück
120 Stk:	5,00 DM	4,167 Pf
40 Stk:		
15 Stk:	1,00 DM	6,667 Pf
3 Stk:	0,25 DM	8,333 Pf
1 Stk:		

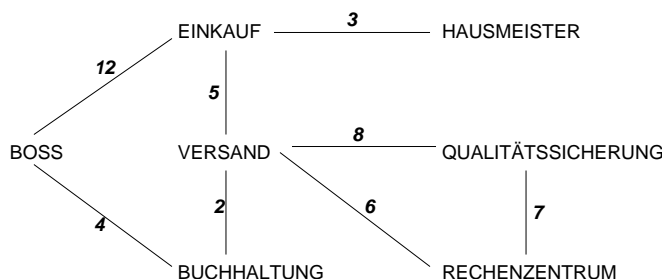
Punkte	
--------	--

Name:

Matrikelnummer:

Aufgabe 3: Graphalgorithmen**(13 Punkte)****(a) (1 Punkt)** Welches Problem löst der Floyd-Warshall-Algorithmus?

(b) (10 Punkt) In einer Firma wird erhoben, wie hoch der Aufwand an Kommunikation von Informationen zwischen verschiedenen Abteilungen/Personen ist. Die Kommunikationskosten setzen sich dabei aus mehreren Aspekten (Erreichbarkeit, Ausmaß an gemeinsamer Vorinformation, etc.) zusammen. Aus Effizienzgründen wurden die Kommunikationskosten nur für ausgewählte Paare von Abteilungen/Personen erhoben. Das Ergebnis der Analyse ist im folgenden Graph abgebildet:



Implementieren Sie den Dijkstra-Algorithmus (in MODULA-2) um zu ermitteln, wie hoch die minimalen Kommunikationskosten für eine fest vorgegebene Person/Abteilung mit allen anderen Personen/Abteilungen sind.

Vorgaben:

```

CONST MatrixDimension = 7;
    UNENDLICH = 99;
  
```

```

TYPE Matrix = ARRAY[1..MatrixDimension],[1..MatrixDimension] OF INTEGER;
    ErgebnisArray = ARRAY[1..MatrixDimension] OF INTEGER;
  
```

Die Knotenbezeichner (“Boss”, “Versand”, etc.) werden dabei durch Zahlen $\{1 \dots MatrixDimension\}$ angegeben. Die Kosten sind als INTEGER vorgegeben.

Gesucht ist die Prozedur PROCEDURE Dijkstra(VAR kMatrix: Matrix; VAR Ergebnis : ErgebnisArray; startKnoten: INTEGER); wobei kMatrix bereits initialisiert sei (Adjazenzmatrix des Graphen).

Punkte	
--------	--

Name:

Matrikelnummer:

Punkte	
--------	--

Name:

Matrikelnummer:

(c) (1 Punkt) Wie hoch sind die minimalen Kommunikationskosten vom "Boss" zu allen anderen Personen/Abteilungen?

(d) (1 Punkt) Was bedeutet "spannender Wald" eines Graphen?

Punkte	
--------	--

Name:

Matrikelnummer:

Aufgabe 4: Dynamische Datenstrukturen

(12 Punkte)

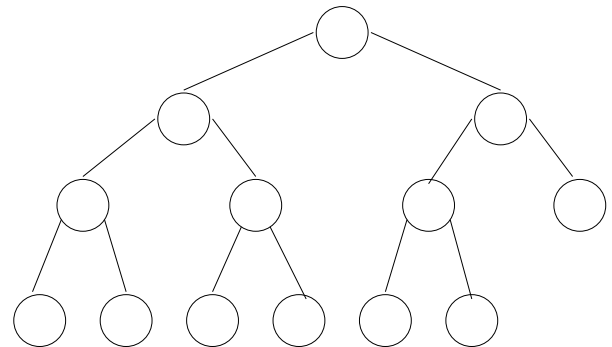
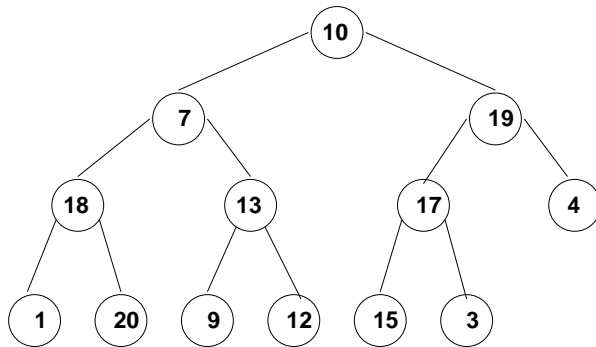
(a) (2 Punkte) Was ist das invariante Ordnungskriterium für geordnete Binärbäume?

Punkte	
--------	--

Name:

Matrikelnummer:

- (b) (3 Punkte)** Bringen Sie für den folgenden Suchbaum (angegeben sind nur die Schlüssel) die Schlüssel in eine korrekte Ordnung. Der resultierende geordnete Binärbaum soll dabei die selbe Struktur aufweisen, wie der gegebene Baum (Lösung in den Musterbaum eintragen!).



- (c) (1 Punkt)** Was ist ein gewichts-balancierter Baum?

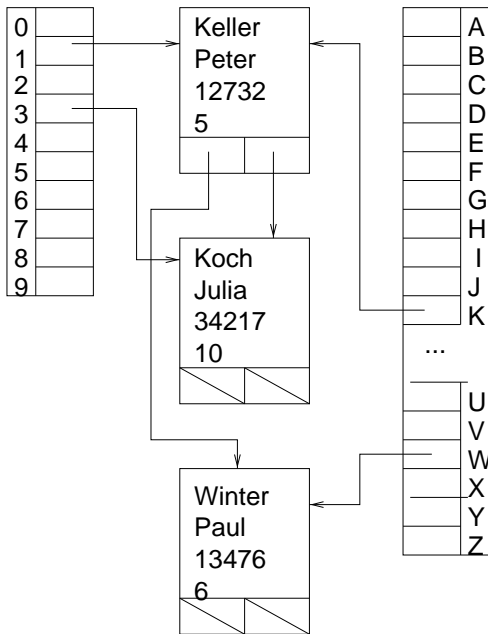
- (d) (1 Punkt)** Was ist doppeltes Hashing?

Punkte	
--------	--

Name:

Matrikelnummer:

(e) (3 Punkte) Die Verwaltung einer Studierenden-Kartei soll optimiert werden. Zu diesem Zweck soll offenes Hashing verwendet werden. Jede "Karteikarte" ist nur einmal gespeichert. Um die Information über Studierende möglichst flexibel abrufbar zu machen, sollen zwei Schlüsseltypen – Matrikelnummer und Nachname – verwendet werden. Das heißt, die Informationen sind über zwei verschiedene Hash-Tabellen abrufbar. Dabei wird die Matrikelnummer als `ARRAY [1..5] OF ziffer`, der Nachname als `String` realisiert. Weitere Informationen seien Vornamen (`String`) und Semesterzahl (`CARDINAL`). Ein Beispiel für die gewünschte "doppelte Verkettung" ist in der folgenden Abbildung gegeben.



Die Information für den Studenten Peter Keller ist über 1 (erste Ziffer der Matrikelnummer) und über *K* (erster Buchstabe des Nachnamens) zugreifbar. Paul Winters Matrikelnummer beginnt ebenfalls mit der Ziffer 1. Seine Informationen sind also in derselben, Hash-Position 1 zugeordneten, Liste zu finden. Julia Kochs Nachname beginnt mit dem Buchstaben *K*. Ihre Informationen sind also in der Liste, die Hash-Position *K* zugeordnet ist, zu finden.

Entwerfen Sie eine geeignete Datenstruktur (Notation in MODULA-2).

Punkte	
--------	--

Name:

Matrikelnummer:

(f) (2 Punkte) Nun sei eine Studierenden-Kartei durch *geschlossenes* Hashing realisiert. Als Schlüssel wird die Summe der Ziffern (0..9) der 5-stelligen Matrikelnummer (Quersumme) verwendet. Als Hashfunktion wird die Divisionsmethode – $h'(k) = k \bmod N$ – verwendet. Zur besseren Übersicht betrachten wir eine Mini-Tabelle mit $N = 13$ Indexplätzen. Gegeben sind der Tabellen-Index j sowie Schlüssel k . Leere Speicherplätze sind durchgestrichen.

j	h'(k)	k
0		
1		
2		15
3		28
4		
5		
6		19
7		
8		
9		9
10		22
11		35
12		

- Tragen Sie $h'(k)$ für alle gegebenen Schlüssel in die Tabelle ein.
- Gesucht wird nun die Information zur Matrikel-Nummer 75995. Als Methode wird lineares Sondieren verwendet.
An welcher Tabellen-Position j wird mit der Suche begonnen?
Geben Sie die Folge der Tabellen-Positionen an, die betrachtet werden, bis der gesuchte Schlüssel gefunden ist.

Punkte	
--------	--