

Name:

Matrikelnummer:

Klausur Informatik B Oktober 1997

Teil I: Informatik 3

Punkte	
--------	--

Name:

Matrikelnummer:

Aufgabe 1: Grundlagen

(je Teilfrage 1 Punkt, gesamt 5 Punkte)

Beantworten Sie folgende Fragen *kurz* in ein oder zwei Sätzen.

- (a) Welche Voraussetzung muß erfüllt sein, damit die „Strategie der lokalen Optimierung“ (*greedy algorithms*) tatsächlich ein globales Optimum ermittelt?**
- (b) Was bedeutet „Grad eines Knotens“ bei Graphen?**
- (c) Was sind „Adjazenzlisten“?**
- (d) Was ist „offenes Hashing“?**
- (e) Beschreibe kurz das Vorgehen bei „globaler Optimierung mit Breitensuche mit bewerteten Kanten“ (*Branch-and-Bound*)?**

Punkte	
--------	--

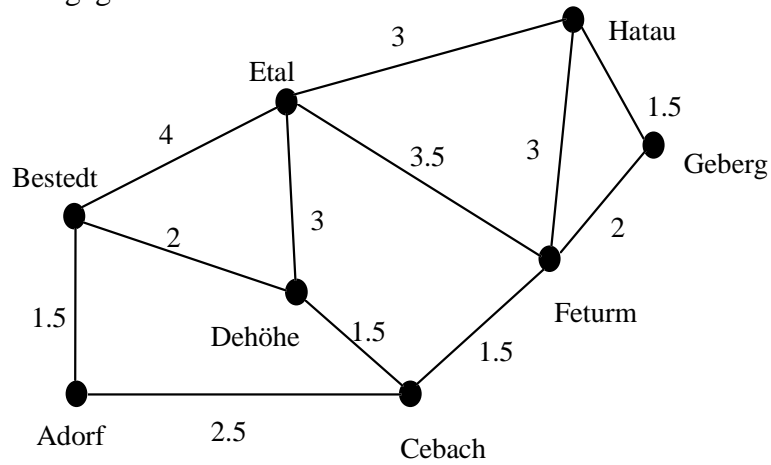
Name:

Matrikelnummer:

Aufgabe 2:

(gesamt 10 Punkte)

Folgende Straßenkarte ist gegeben:



- a) Finden Sie mit dem A*-Algorithmus den kürzesten Weg von Adorf nach Hatau! Verwenden Sie dazu folgende untere Schranken für die Reststrecke: **(8 Punkte)**

von ... nach Hatau	Adorf	Bestedt	Cebach	Dehöhe	Etal	Feturm	Geberg	Hatau
h*	4	3.5	3	3	2	2	1	0

- b) Haben Sie mit dieser Schätzung eine global optimale Lösung gefunden? **(Begründung)**

(2 Punkte)

Punkte	
--------	--

*Name:**Matrikelnummer:*

Aufgabe 3:**(10 Punkte)**

Schreiben Sie ein Modula-Programm, das mit dem Floyd-Warshall-Algorithmus alle kürzesten Wege zwischen den Orten auf der Karte aus Aufgabe 2 findet!

Punkte	
--------	--

Name:

Matrikelnummer:

Aufgabe 4: Hashing

(gesamt 15 Punkte)

Es sollen eindeutig unterscheidbare 6-stellige Telefonnummern in eine Hashtabelle einsortiert werden. Bisher gibt es drei Telefonnummern:

Telefonnummer	Name
432 107	Mrs. Money Penny
498 507	Dr. No
111 007	James Bond

Es werden nicht mehr als 100 Telefonnummern erwartet, daher wird die Größe der Tabelle auf **N = 100** festgelegt.

(a) **Welche Eigenschaften sollen Hashfunktionen besitzen? (Nennen Sie mindestens 2) (2 Punkte)**

(b) **Finden Sie eine Hashfunktion hash, welche die unter (a) aufgeführten Eigenschaften besitzt, und sortieren Sie die Kunden in die Tabelle ein. Dabei soll es für die drei gegebenen Datensätze zu keiner Kollision kommen! Geben Sie die Hashfunktion in MODULA-2 an und beschreiben Sie kurz die Idee der Hashfunktion. (2 Punkte)**

Hashschlüssel	Telefonnummer	Name
:	:	:
:	:	:
:	:	:
:	:	:

```
PROCEDURE hash (nummer : LONGCARD): CARDINAL;
(* Liefert einen Hashschlüssel zu einer vorgegebenen
   Telefonnummer *)
```

Punkte	
--------	--

*Name:**Matrikelnummer:*

- (c) Finden Sie nun eine neue Telefonnummer, deren Hashwert mit einer der drei ersten Nummern kollidiert. **(1 Punkte)**

- (d) Welche Möglichkeiten gibt es, diese Kollision zu behandeln? Wählen Sie eine aus, beschreiben Sie stichpunktartig, wie sie funktioniert, und implementieren Sie die Suchfunktion "Search" in MODULA-2. Dazu müssen Sie die vorgegebene Typdefinitionen ergänzen. **(10 Punkte)**

```
TYPE t_data = RECORD
    nummer: LONGCARD;
    name: STRING;
END;
t_hashtable = ARRAY [1..100] OF t_my_hash_type;
```

```
VAR table: t_hashtable;
```

```
PROCEDURE Search (schlüssel: CARDINAL; VAR data: t_data): BOOLEAN;
(* Liefert Datensatz mit Schlüssel "schlüssel" und TRUE zurück,
falls in der Tabelle vorhanden. Sonst wird FALSE geliefert. *)
```

Punkte	
--------	--

*Name:**Matrikelnummer:*

Aufgabe 5: Binär-Bäume**(10 Punkte)**

Folgende Typdefinition für einen **geordneten Binär-Baum** sei gegeben:

```
TYPE t_baum_ptr = POINTER TO t_baum;

TYPE t_data_ptr = POINTER TO t_data;

TYPE t_blatt = RECORD
    schluessel : CARDINAL;
    data : t_data_ptr;
END;

TYPE t_baum = RECORD
    vorgaenger : t_baum_ptr;
    CASE typ: (zweig_t, blatt_t) OF
        blatt_t : blatt : t_blatt;
        | zweig_t : links : t_baum_ptr;
                    schluessel : CARDINAL;
                    rechts : t_baum_ptr;
    END (*CASE*)
END; (*RECORD*)
```

Hinweis: Beim Typ `t_baum` handelt es sich um einen sogenannten varianten-RECORD. In Abhängigkeit des Werts des Recordelements `typ` (`zweig` oder `blatt`) sind die jeweiligen anderen Elemente des Records ansprechbar. Beispiel: Ist `typ = zweig` kann man die Elemente `links`, `schluessel` und `rechts` benutzen, `blatt` aber nicht. Für die Wurzel des Baumes gilt `vorgaenger = NIL`.

Punkte	
--------	--

*Name:**Matrikelnummer:*

Implementieren sie folgende Prozedur in MODULA-2:

```
PROCEDURE loeschen (VAR baum : t_baum_ptr, blatt : t_blatt );
  (* Loescht am gegebenen Baum "baum" ein Blatt "blatt".
   Dabei sollen KEINE Operationen zur Ausgewogenheit
   durchgefuehrt werden. Falls es kein Blatt mit dem gesuchten
   Wert fuer "schluessel" gibt, soll eine Fehlermeldung
   ausgegeben werden. *)
```

Punkte	
--------	--