Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

09.04.2013
Sommersemester 2013

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK II
BLATT 1

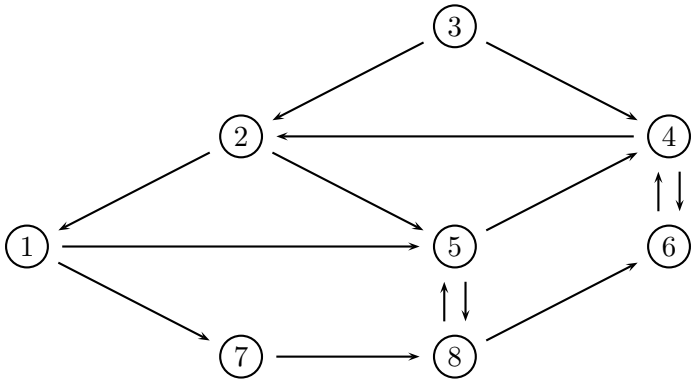
(1) Gegeben sei der gerichtete Graph $G = (V, E)$, wobei

$$V = \{1, 2, 3, 4, 5, 6\},$$
$$E = \{(1,3), (1,4), (2,1), (2,3), (2,4), (3,4), (3,6), (4,5), (5,2), (6,5)\}.$$

Skizzieren Sie den Graphen. Geben Sie für jedes Element $v \in V$ den Ausgangs- und Eingangsgrad an. Stellen Sie den Graphen in einer Adjazenz- und Inzidenzmatrix dar.

(2) Beweisen Sie: Zwischen Knoten eines Graphen gibt es genau dann einen Weg, wenn es einen Pfad zwischen ihnen gibt.

(3) ★ Gegeben sei der folgende gerichtete Graph G.



(a) Geben Sie für den Graphen G die Adjazenzmatrix A und die Inzidenzmatrix B an.
(b) Führen Sie eine Breitensuche für den Graphen G ausgehend von Knoten 1 durch. Gehen Sie dabei wie in Beispiel 5.11 der Vorlesung vor, d.h. skizzieren Sie für jeden Schleifendurchlauf den Graphen unter Angabe der Farbe, der Abstandsschätzung und, falls gefunden, der Vorgänger aller Knoten. Geben Sie auch immer die Warteschlange an.

(4) ★ Schreiben Sie ein Programm, mit dessen Hilfe ein gerichteter Graph mit k Knoten als Adjazenzmatrix gespeichert werden kann. Die Anzahl der Knoten soll vom Benutzer zu Beginn dynamisch eingegeben werden können. Anschließend soll der Benutzer die Möglichkeit bekommen, zu jedem j-ten Knoten, $1 \leq j \leq k$, nacheinander die adjazenten Knoten einzugeben. Die Ausgabe des Programms ist die Adjazenzmatrix.

Hinweis. Speichern Sie die Adjazenzmatrix in einem dynamischen `array` der Länge k^2.

(5) Schreiben Sie ein Programm, mit dessen Hilfe ein gerichteter Graph mit beliebig vielen Knoten als Adjazenzliste gespeichert werden kann. Speichern Sie dazu die eingegebenen Knoten in einer Liste, deren Nutzdaten auch Listen sind, in denen die jeweiligen

adjazenten Knoten abgespeichert werden sollen.
Um diese Aufgabe zu bearbeiten, haben Sie bis zum 23.04.13 Zeit.

Hinweis. Diese Programm benötigt das Konzept „Listen von Listen“. Dieses wird in den Übungen am 10./15.4. besprochen.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen: 16.04.2013 vor der Vorlesung

---

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben:
Grundsätzlich müssen Sie Programme per E-mail abgeben. Weitere Hinweise und genauere Informationen zur Abgabe (E-mail-Adresse) finden Sie in Abschnitt 5 des Leitfadens zur Vorlesung:

http://www-ian.math.uni-magdeburg.de/∼pulst/Algo_Math_II/Leitfaden.pdf.

Ergänzend zur E-mail-Abgabe geben Sie bitte eine handschriftliche Lösung ab. Diese muss Kernelemente des Programms angemessen dokumentieren; Standardinitialisierungs-, Ein- und Ausgabefunktionen können – analog zur Vorlesung – summarisch dargestellt werden.

Die Beispielprogramme von Prof. Grunau finden Sie unter
http://www-ian.math.uni-magdeburg.de/home/grunau/additional.html.

Die Aufgabenblätter und Programme aus der Übung finden Sie unter
http://www-ian.math.uni-magdeburg.de/∼pulst/Algo_Math_II.html.

Hinweise zur Installation einer Programmierumgebung für die Programmiersprache C sind in dem Leitfaden zur Vorlesung beschrieben:

http://www-ian.math.uni-magdeburg.de/∼pulst/Algo_Math_II/Leitfaden.pdf.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

16.04.2013
Sommersemester 2013

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK II
BLATT 2

(1) Sei G ein gerichteter Graph mit folgender Eigenschaft: Zwischen jedem Paar von Knoten v, w gibt es entweder die Kante (v, w) oder die Kante (w, v). Beweisen Sie, dass der Graph genau dann einen geschlossenen Weg besitzt, wenn er einen geschlossenen Weg der Länge 3 besitzt.

(2) ★ Sei G ein gerichteter Graph mit Adjazenzmatrix A. Sei $A^k = \overbrace{A \cdot \ldots \cdot A}^{k\text{-mal}}$ die k-te Potenz von A, $k \in \mathbb{N}$. Beweisen Sie per vollständiger Induktion: Der (i, j)-te Eintrag von A^k ist gleich der Anzahl der verschiedenen Pfade mit Anfangsknoten i und Endknoten j, welche aus k Kanten bestehen.

(3) ★ *(Aufgabe 5, Blatt 1)*
Schreiben Sie ein Programm, mit dessen Hilfe ein gerichteter Graph mit beliebig vielen Knoten als Adjazenzliste gespeichert werden kann. Speichern Sie dazu die eingegebenen Knoten in einer Liste, deren Nutzdaten auch Listen sind, in denen die jeweiligen adjazenten Knoten abgespeichert werden sollen.

Hinweis. Diese Programm benötigt das Konzept „Listen von Listen“.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben. Abgabe der Lösungen: 23.04.2013 vor der Vorlesung.

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

23.04.2013
Sommersemester 2013

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK II
BLATT 3

ACHTUNG! Der Abgabezeitpunkt hat sich geändert. Genauere Informationen finden Sie am Ende dieses Blattes.

(1) Sei $G = (V, E)$ ein gerichteter Graph. Wir führen auf V eine Erreichbarkeitsrelation ein: $u \sim v$ genau dann, wenn es einen Pfad von u nach v gibt. Nullpfade (v) für $v \in V$ sind ausdrücklich zugelassen und gelten auch als Zyklen. Zeigen Sie die Äquivalenz der folgenden Aussagen:

(a) „$\sim$“ ist eine Äquivalenzrelation.

(b) Für je zwei Knoten $u, v \in V$ impliziert $u \sim v$ stets, dass es einen Zyklus gibt, der u und v enthält.

(c) $\forall u, v \in V : \quad d(u, v) = \infty \Leftrightarrow d(v, u) = \infty$.

In diesem Falle nennt man die erzeugten Äquivalenzklassen *Zusammenhangskomponenten*; falls V aus nur einer Zusammenhangskomponente besteht, nennt man V zusammenhängend.

(2) ★ Sei $G = (V, E)$ ein gerichteter Graph mit Knotenmenge $V = \{1, \ldots, k\}$ und Adjazenzmatrix A. Wir setzen voraus, dass die Erreichbarkeitsrelation „$\sim$“ aus der vorhergehenden Aufgabe eine Äquivalenzrelation ist. Sei $A^j = \overbrace{A \cdot \ldots \cdot A}^{j\text{-mal}}$ die j-te Potenz von A,

$$B = \sum_{j=0}^{k-1} A^j.$$

Zeigen Sie: Eine Teilmenge $\emptyset \neq M$, $M \subset V$ ist genau dann eine Zusammenhangskomponente von V, falls gilt:

$$\forall u, v \in M : \quad b_{uv} > 0 \quad \text{und} \quad \forall u \in M, v \in V \setminus M : \quad b_{uv} = b_{vu} = 0.$$

(3) Entwickeln Sie das Programm `Breitensuche.c` aus der Vorlesung dahingehend weiter, dass Sie die Warteschlange als Liste programmieren.

(4) ★ Entwickeln Sie das Programm `Breitensuche.c` aus der Vorlesung unter Verwendung des Programmes aus Aufgabe 4, Blatt 1 weiter:

- Wie in der vorherigen Aufgabe sollen Sie die Warteschlange als Liste programmieren.
- Weitergehend sollen die verwendeten arrays alle dynamisch angelegt werden, nachdem der Benutzer zu Beginn des Programms die Zahl k der Knoten eingegeben hat.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben.

ACHTUNG: *Die Abgabe der Lösungen (handschriftlich und per E-mail) für* **dieses** *Blatt ist am*

Donnerstag, den 02.05.13 um 11.00 Uhr.

Reichen Sie Ihre handschriftlichen Lösungen bitte vor Raum **G02-06** *ein. (Handschriftliche Lösungen können Sie auch schon vor der Vorlesung am Dienstag, den 30.04.13, abgeben.)*

Für die **Mittwochsgruppe** *findet eine Übung am Donnerstag, den 02.05.13, von 13.00 Uhr - 15.00 Uhr im Raum G02-112 statt. Die Übung am Mittwoch, den 15.05.13, entfällt.*

Abgabe ab dem **nächsten** *Blatt: Ab Blatt 4 findet die Abgabe (handschriftlich und per E-mail) immer am darauffolgenden Mittwoch um 11.00 Uhr statt. Abgabe der handschriftlichen Lösung ist vor Raum G02-06.*

Der Klausurtermin steht fest:

Freitag, den 05.07.13, 15.15 Uhr - 16.45 Uhr, im Raum G03-315.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

30.04.2013
Sommersemester 2013

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK II
BLATT 4

(1) Sei $G = (V, E)$ ein gerichteter Graph mit Knotenmenge $V = \{1, \ldots, k\}$ und Adjazenzmatrix A. Auf der Internetseite von Prof. Grunau finden Sie das Programm `Abstandsbestimmung.c` – eine Variante des Programms `Breitensuche.c` aus der Vorlesung. Dieses Programm bestimmt $d(u, v)$ für alle $u, v \in V$. Entwickeln Sie dieses Programm wie folgt weiter:

(a) Das Programm soll zunächst prüfen, ob es sich bei der Erreichbarkeitsrelation „$\sim$“ aus Blatt 3, Aufgabe 1 um eine Äquivalenzrelation handelt.

(b) ★ Falls dieses der Fall ist, soll das Programm die Zusammenhangskomponenten von V ermitteln und ausgeben.

Die Aufgaben 1 und 2 von Blatt 3 erlauben es, die gestellte Aufgabe sowohl durch die Bestimmung der Abstandsmatrix $(d(u, v))_{u,v=1,\ldots,k}$ als auch durch die Bestimmung der „Erreichbarkeitsmatrix“ $B = \sum_{j=0}^{k-1} A^j$ zu lösen. Bevor Sie sich für eine Strategie entscheiden, führen Sie eine bitte Laufzeitanalyse durch und wählen Sie die (asymptotisch) schnellere.

(2) ★ Sei $G = (V, E)$ ein ungerichteter Graph ohne Schlingen mit Knotenmenge $V = \{1, \ldots, k\}$ und Adjazenzmatrix A, d.h. die Adjazenzmatrix A ist symmetrisch und die Erreichbarkeitsrelation „$\sim$“ aus Blatt 3, Aufgabe 1 ist sicher eine Äquivalenzrelation. Es bezeichne $m = \frac{1}{2}\#E$, von den Kanten $(u, v) \in E \Leftrightarrow (v, u) \in E$ wird also jeweils nur eine gezählt.

(a) Zeigen Sie: Gilt $m > \frac{1}{2}(k-1)(k-2)$, so ist V zusammenhängend.

(b) Gibt es ungerichtete Graphen mit $m = \frac{1}{2}(k-1)(k-2)$, die nicht zusammenhängend sind? Wenn ja, geben Sie Beispiele an.

(3) (a) Entwickeln Sie das Programm aus Blatt 2, Aufgabe 3 zur Eingabe von Listen dahingehend weiter, dass Sie nun für jeden Knoten des Graphen einen Knoten in Ihrer Adjazenzliste anlegen. Fangen Sie unzulässige Eingaben ab und stellen Sie sicher, dass Ihre Adjazenzliste einen Graphen ohne Schlingen darstellt. Ihre Eingabefunktion soll die Zahl der Knoten zurückgeben.

(b) In einem zweiten Schritt:
- sollen Sie die Deklaration von `OberKnoten` dahingehend erweitern, dass Sie die für die Breitensuche erforderlichen Attribute „Abstand, Farbe, Vorgänger“ einfügen;
- soll der Nutzer nach einem Startknoten gefragt werden;
- soll dieser zusammen mit dem Listenkopf an eine Funktion `Initialisiere` übergeben werden, die die Attribute entsprechend des Initialisierungsschrittes des Breitensuchealgorithmus setzt;
- sollen Sie eine zweite Ausgabefunktion für die Adjazenzliste vorbereiten, in der Erreichbarkeit, Abstand und Vorgängerkette ausgegeben werden.

(4) ★ Entwickeln Sie das in der vorhergehenden Aufgabe entwickelte Programm und das Breitensuche-Programm aus Blatt 3, Aufgabe 4 dahingehend weiter, dass Sie nun ganz auf arrays verzichten und diese vollständig durch Listen ersetzen.

Hinweis. Dieses Programm müssen Sie erst am Mittwoch, den 15.05.2013 abgeben.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben.

Abgabe der Lösungen: Mittwoch, den 08.05.2013 bis 11.00 Uhr vor Raum 02-06 (Büro L. Pulst).

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

07.05.2013
Sommersemester 2013

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK II
BLATT 5

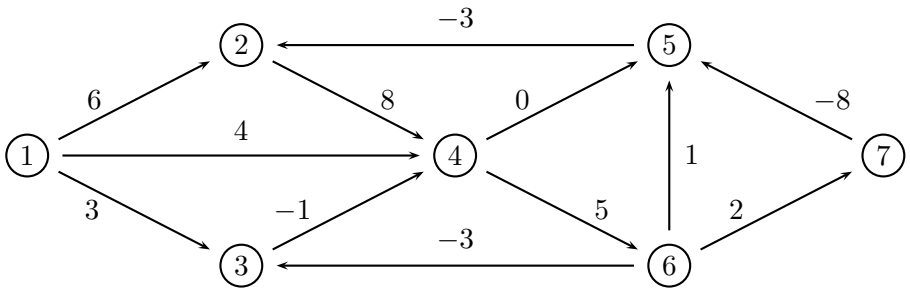
(1) (a) Fügen Sie in das in Blatt 4, Aufgabe 3 entwickelte Programm eine Funktion

```
struct OberKnoten * Knotenfinder(struct Listenkopf *kopf, short nummer)
```

ein, die die Knotenadresse zu einer eingegebenen Knotennummer findet.

(b) ★ *(Aufgabe 4, Blatt 4)*
Entwickeln Sie das vorhergehende Programm und das Breitesuche–Programm aus Blatt 3, Aufgabe 4 dahingehend weiter, dass Sie nun ganz auf arrays verzichten und diese vollständig durch Listen ersetzen.

(2) ★ Gegeben sei der folgende gerichtete gewichtete Graph G.



Führen Sie den Bellman-Ford-Algorithmus mit Startknoten 1 durch. Skizzieren Sie die Initialisierung sowie den Graphen G in den Hauptschleifendurchläufen immer dann, wenn sich durch die Funktion `Relax(u,v)` für $(u, v) \in E$ etwas ändert. Benutzen Sie für die Skizzen die „Notation“ aus Beispiel 6.16.

(3) Sei $G = (V, E)$ ein gewichteter gerichteter Graph mit der Gewichtsfunktion $W : E \to \mathbb{R}$ und einer Funktion $h : V \to \mathbb{R}$, die die Knotenmenge auf die Menge der reellen Zahlen abbildet. Für jede Kante $(u, v) \in E$ definieren wir

$$W'(u, v) = W(u, v) + h(u) - h(v).$$

Beweisen Sie:

(a) Genau dann hat ein Pfad von u nach v in G minimales Gewicht bezüglich W, wenn er minimales Gewicht bezüglich W' hat.

(b) G besitzt unter der Gewichtsfunktion W genau dann einen Zyklus mit strikt negativem Gewicht, wenn G unter der Gewichtsfunktion W' einen Zyklus mit strikt negativem Gewicht besitzt.

(c) Der Graph G besitze nun keinen Zyklus mit strikt negativem Gewicht. Konstruieren Sie eine Funktion $h : V \to \mathbb{R}$, so dass alle Kanten in E bezüglich der Gewichtsfunktion W' nichtnegatives Gewicht haben.

Hinweis. Betrachten Sie für $s \notin V$ den Graphen $G' = (V', E')$ mit $V' = V \cup \{s\}$, $E' = E \cup \{(s, v) : v \in V\}$ und $W(s, v) := 0$ für alle $v \in V$.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben.

Abgabe der Lösungen: Mittwoch, den 15.05.2013 bis 11.00 Uhr vor Raum 02-06 (Büro L. Pulst).

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

14.05.2013
Sommersemester 2013

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK II
BLATT 6

(1) ★ In dieser Aufgabe soll der Bellman-Ford-Algorithmus mit Listen programmiert werden.

(a) Verändern Sie das Programm aus Aufgabe 1, Blatt 5, dahingehend, dass die Strukturdefinitionen und Funktionen für den Bellman-Ford-Algorithmus vorbereitet werden. Insbesondere müssen die `Unterknoten` ein Nutzdatum `Gewicht` erhalten, dieses soll in der Funktion `Eingeben` erfragt werden. Lassen Sie sich auch das betragsmäßig größte eingegebene Gewicht über einen Zeiger von der Funktion `Eingeben` zurückgeben, dieses wird für die Setzung von `UNENDLICH` benötigt.

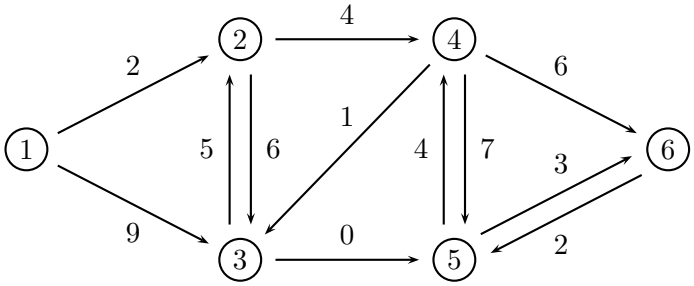
(b) Erweitern Sie das Programm aus der vorherigen Aufgabe um den Bellman-Ford-Algorithmus.

(2) ★ Konstruieren Sie einen gerichteten gewichteten Graphen $G = (V, E)$ mit $k \in \mathbb{N}$ Knoten, d.h. $V = \{1, 2, \ldots, k\}$, so dass der Bellman-Ford-Algorithmus die $(k-1)$ Hauptschleifendurchläufe für einen Startknoten $s \in V$ auch benötigt, d.h. bei Anwendung des Algorithmus soll die Relaxationsprozedur `Relax(u,v)`, $u, v \in V$, auch im $(k-1)$. Hauptschleifendurchlauf die Abstandsschätzung eines Knotens verändern.

Für die folgenden Aufgaben benötigen Sie den Stoff aus der Vorlesung vom 21.05.13.

(3) Sei $G = (V, E)$ mit $V = \{1, 2, \ldots, k\}, k \geq 2$, ein gerichteter gewichteter Graph ohne Schlingen und Gewichtsfunktion $W : E \to [0, \infty)$. Zeigen Sie, dass die Laufzeit des Dijkstra-Algorithmus bei festem Startknoten $O(k^2)$ beträgt.

(4) ★ Gegeben sei der folgende gerichtete gewichtete Graph G.



Führen Sie den Dijkstra-Algorithmus mit Startknoten 1 durch. Skizzieren Sie die Initialisierung sowie den Graphen G in jedem Hauptschleifendurchlauf. Benutzen Sie für die Skizzen die „Notation“ aus Beispiel 6.22.

(5) Sei $G = (V, E)$ ein gerichteter gewichteter Graph ohne Schlingen mit strikt positiver Gewichtsfunktion $W : E \to \{1, 2, \ldots, M\}$, wobei $M \in \mathbb{N}$. Für alle Knoten $u, v \in V, u \neq v$ gilt: ist $(s, \ldots, u)$ ein kürzester Pfad vom Startknoten s nach u und $(s, \ldots, v)$ ein kürzester Pfad von s nach v dann ist $W(s, \ldots, u) \neq W(s, \ldots, v)$. Wir definieren

nun einen ungewichteten gerichteten Graphen $G' = (V \cup V', E')$, indem wir jede Kante $(u, v) \in E$ durch $W(u, v)$ aufeinanderfolgende Kanten vom Gewicht 1 ersetzen.

(a) Berechnen Sie die Anzahl der Knoten von G'.

(b) Wir wenden nun die Breitensuche auf G' an. Zeigen Sie, dass die Reihenfolge, in der die Breitensuche auf G' die Knoten aus V schwarz färbt, die gleiche ist, in der Dijkstras Algorithmus angewendet auf G die Knoten aus V aus der Warteschlange (des Dijkstra-Algorithmus) extrahiert.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben.

Abgabe der Lösungen: Mittwoch, den **29.05.2013** *bis 11.00 Uhr vor Raum 02-06 (Büro L. Pulst).*

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Bitte beachten Sie: die Übungen am Mittwoch, den 15.05.13, und Montag, den 20.05.13, entfallen.

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

28.05.2013
Sommersemester 2013

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK II
BLATT 7

(1) ★ Programmieren Sie den Dijkstra-Algorithmus mittels Listen.

Hinweis. Bitte bauen Sie auf der Musterlösung zu Aufgabe 6.1 auf. Ob ein Knoten aktuell zu der Menge Q bzw. deren Komplement $S = V \setminus Q$ gehört, kodieren Sie bitte über eine Attribut `Farbe` in `struct OberKnoten`, das Sie entsprechend initialisieren bzw. aktualisieren müssen. Im wesentlichen besteht die Aufgabe nun aus dem Schreiben einer Funktion

```
struct OberKnoten *ExtractMinQ(struct Listenkopf *kopf, short unend)
```

sowie dem Anpassen der Hauptschleife an den Dijkstra-Algorithmus. Es reicht, wenn Sie dieses in der handschriftlichen Abgabe dokumentieren.

(2) ★ Betrachten Sie die Sinusreihe $x \mapsto \sin x = \sum_{k=0}^{\infty} \frac{(-1)^k}{(2k+1)!} x^{2k+1}$ und das approximierende Taylorpolynom $T_K(x) := \sum_{k=0}^{K} \frac{(-1)^k}{(2k+1)!} x^{2k+1}$ vom Grade $(2K+1)$.

(a) Bestimmen Sie K so, dass $\sup_{x\in[-2,2]} |T_K(x) - \sin x| \leq 10^{-15}$.

(b) Implementieren Sie mit diesem K diese Approximation T_k in einem C-Programm und vergleichen Sie mit der vorimplementierten Sinusfunktion. Werden Ihre Erwartungen erfüllt?

(c) Ergänzen Sie das vorhergehende Programm um `gnuplot`-Visualisierungen auf vom Benutzer einzugebenden Bereichen. Vergleichen Sie dazu bitte das Beispielprogramm `Exponentialfunktion2.c`.
Geben Sie bitte Ausdrucke für die Intervalle $[-2, 2]$ und $[0, 12]$ ab.

(d) Zeigen Sie, dass für alle $K \in \mathbb{N}_0$ gilt: $\sup_{x\in\mathbb{R}} |T_K(x) - \sin x| = \infty$.

(3) Sei $G \subset \mathbb{R}^2$ ein Gebiet. Für Funktionen $u : G \to \mathbb{R}$ definiert man den Laplaceoperator durch $\Delta u(x, y) = \frac{\partial^2}{\partial x^2} u(x, y) + \frac{\partial^2}{\partial y^2} u(x, y)$. Stehende Wellen werden durch das Eigenwertproblem

$$-\Delta u(x, y) = \lambda u(x, y) \text{ in } G \tag{1}$$

zusammen mit geeigneten Randbedingungen auf ∂G modelliert. Der Eigenwertparameter $\lambda > 0$ ist ebenfalls Teil des Problems und muss bestimmt werden. Allerdings nehmen wir zunächst der Einfachheit halber an, dass $G = \mathbb{R}^2$.

(a) Begründen Sie, dass man in dieser speziellen Situation o.B.d.A. $\lambda = 1$ annehmen kann. Fortan sei also stets $G = \mathbb{R}^2$, $\lambda = 1$.

(b) Wir vereinfachen die Situation weiter und suchen nur radialsymmetrische Lösungen in der Form $u(x, y) = \tilde{u}(|(x, y)|)$. Welche Gleichung für $\tilde{u}$ ist in diesem Falle äquivalent zu (1)?

(c) Machen Sie den Ansatz $\tilde{u} = \sum_{k=0}^{\infty} a_k r^{2k}$ und leiten Sie durch Koeffizientenvergleich eine Rekursionsformel für die a_k her, so dass $\tilde{u}$ die in (b) hergeleitete

radialsymmetrische Version von (1) löst. Wieso kann man zunächst $a_0 = 1$ annehmen?

(d) Diskutieren Sie das Konvergenzverhalten der in (c) erhaltenen Reihe $\tilde{u}$ und begründen Sie rigoros, dass damit $u(x, y) := \tilde{u}(|(x, y)|)$ in der Tat (1) löst.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben.

Abgabe der Lösungen: Mittwoch, den 05.06.2013 bis 11.00 Uhr vor Raum 02-06 (Büro L. Pulst).

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Bitte beachten Sie die folgenden Informationen zur Klausur:

- *Die Klausur findet am Freitag, den 05.07.2013 in Raum G03-315 von 15.15 Uhr - 16.45 Uhr (d.h. Dauer 90 Minuten) statt. Bitte seien Sie pünktlich da.*
- *Bringen Sie bitte ausreichend Papier, Schreibzeug (Kugelschreiber oder Füller) und einen Lichtbildausweis mit.*
- *Während der Klausur dürfen Sie das* **gedruckte** *Vorlesungsskript von Prof. Grunau und ein festgebundenes Programmierhandbuch ihrer Wahl benutzen, allerdings keine losen Blätter-Sammlungen.*
- *Es sind keinerlei elektronische Hilfsmittel (Smartphones, Laptops, Netbooks, etc.) erlaubt.*

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

04.06.2013
Sommersemester 2013

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK II
BLATT 8

(1) ★ Betrachten Sie wie in Aufgabe 7.3 die Besselfunktion

$$J_0(r) = \sum_{k=0}^{\infty} \frac{(-1)^k}{4^k (k!)^2} r^{2k}.$$

Bestimmen und implementieren Sie eine Approximation für $J_0(r)$, die für $r \in [0, 4]$ unter Berücksichtigung der Rundungsfehler maximal um 10^{-10} von $J_0(r)$ abweicht. Visualisieren Sie diese Approximation graphisch analog zu den Beispielprogrammen aus der Vorlesung und von Übungsblatt 7.

(2) Sei $f(x) = \sum_{k=0}^{\infty} a_k x^k$ eine nahe $x_0 = 0$ konvergente Potenzreihe mit Konvergenzradius $\rho_f > 0$. Schreiben Sie ein Programm, das es einem Nutzer gestattet, „beliebig viele“ dieser Koeffizienten a_k einzugeben, diese in einer Liste abspeichert und anschließend die entsprechenden Approximationen punktweise bzw. graphisch durchführt. Schreiben Sie eine Variante dieses Programms, das eine vom Nutzer einzugebende Zahl von Koeffizienten etwa der Sinusreihe automatisch berechnet, in einer Liste abspeichert und weiter wie vorher verfährt.

Bemerkung. Diese Aufgabe dient der programmiertechnischen Vorbereitung von Aufgabe 3. Sie soll **nicht** suggerieren, dass diese Berechnungsweise zur Auswertung von Taylorpolynomen großen Grades geeignet sei. Werten Sie z.B. die Sinusreihe mit 501 Gliedern in $x = 5$ mit diesem Programm aus und vergleichen Sie mit dem modifizierten Beispielprogramm von Blatt 7.

(3) ★ Sei $f(x) = \sum_{k=0}^{\infty} a_k x^k$ eine nahe $x_0 = 0$ konvergente Potenzreihe mit Koeffizienten $a_k \in \mathbb{R}$ und Konvergenzradius $\rho_f > 0$. Sicher ist dann f auf $(-\rho_f, \rho_f)$ beliebig oft differenzierbar. Wir nehmen weiter an, dass $a_0 = f(0) \neq 0$ gilt. Wir wollen $g = \frac{1}{f}$ für x nahe 0 betrachten. Sicher ist dort auch g beliebig oft differenzierbar. Das Ziel dieser Aufgabe besteht darin zu zeigen, dass g nahe 0 sogar als konvergente Potenzreihe

$$g(x) = \sum_{k=0}^{\infty} b_k x^k$$

mit geeigneten Koeffizienten b_k dargestellt werden kann, wobei für den Konvergenzradius $\rho_g > 0$ gilt.

(a) Nehmen Sie zunächst an, dass g eine Darstellung als konvergente Potenzreihe gestattet und leiten Sie eine Rekursionsformel für die Koeffizienten b_k her.

(b) Implementieren Sie diese Rekursionsformel. Dabei sollen die Anfänge der Koeffizientenfolgen $(a_k)_{k\in\mathbb{N}_0}$, $(b_k)_{k\in\mathbb{N}_0}$ mit Hilfe von Listen abgespeichert werden. Wie in der vorhergehenden Aufgabe soll der Nutzer zunächst „beliebig viele“ dieser Koeffizienten a_k eingeben; Ihr Programm ermittelt dann die entsprechende Zahl der Koeffizienten b_k.

(c) Zeigen Sie, dass für den Konvergenzradius ρ_g der formal in (a) bestimmten Potenzreihe g in der Tat $\rho_g > 0$ gilt. Folgern Sie damit für $\rho_0 := \min\{\rho_f, \rho_g\}$, dass in der Tat $\forall x \in (-\rho_0, \rho_0) : f(x) \cdot g(x) = 1$ gilt.

Hinweis. Zeigen Sie zunächst, dass Sie o.B.d.A. $a_0 = 1$ annehmen können. Aus der Analysisvorlesung wissen Sie, dass es eine Zahl $M \geq 0$ gibt derart, dass $\forall k \in \mathbb{N} : \quad |a_k| \leq M^k$, wegen $a_0 = 1$ gilt dieses dann auch für $k = 0$. Zeigen Sie induktiv für die in (a) definierten Koeffizienten b_k, dass $|b_k| \leq (2M)^k$ gilt. Folgern Sie daraus $\rho_g \geq \frac{1}{2M}$.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben.

Abgabe der Lösungen: Mittwoch, den 12.06.2013 bis 11.00 Uhr vor Raum 02-06 (Büro L. Pulst).

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Bitte beachten Sie die folgenden Informationen zur Klausur:

- *Die Klausur findet am Freitag, den 05.07.2013 in Raum G03-315 von 15.15 Uhr - 16.45 Uhr (d.h. Dauer 90 Minuten) statt. Bitte seien Sie pünktlich da.*
- *Bringen Sie bitte ausreichend Papier, Schreibzeug (Kugelschreiber oder Füller) und einen Lichtbildausweis mit.*
- *Während der Klausur dürfen Sie das* **gedruckte** *Vorlesungsskript von Prof. Grunau und ein festgebundenes Programmierhandbuch ihrer Wahl benutzen, allerdings keine losen Blätter-Sammlungen.*
- *Es sind keinerlei elektronische Hilfsmittel (Smartphones, Laptops, Netbooks, etc.) erlaubt.*

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

18.06.2013
Sommersemester 2013

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK II
BLATT 9

(1) ★ In dieser Aufgabe soll das Programm aus Aufgabe 8.3 b) modizifiert werden.

(a) Zunächst soll der Nutzer die Koeffizienten a_k von $f(x) = \sum_{k=0}^{\infty} a_k x^k$ als Bruch aus Zähler und Nenner eingeben können. Speichern Sie dazu Zähler und Nenner des Bruchs als zwei `integer`-Zahlen als Nutzdaten in Ihrer Koeffizienten-Liste. Die Koeffizienten der Potenzreihen von $\frac{1}{f}$ sollen dann ebenfalls als Bruch aus Zähler und Nenner berechnet und vollständig gekürzt ausgegeben werden.

Hinweis. Benutzen Sie das ggT-Programm und beachten Sie auch die Bruchrechnungsaufgabe 4.3 aus Algorithmischer Mathematik I.

(b) Verändern Sie die Ausgabe des vorherigen Programms dahingend, dass für den Zähler und den Nenner die jeweilige Primfaktorzerlegung ausgegeben wird.

Hinweis. Benutzen Sie das Programm `Primfaktorzerlegung.c` aus der Vorlesung Algorithmische Mathematik I.

(c) Automatisieren Sie das Programm aus b) für die Exponentialreihe.

(2) ★

(a) Implementieren Sie in einem C-Programm ein Bisektionsverfahren um die Nullstelle der Funktion $x \mapsto \sin^2(x) - \frac{1}{2}$, $x \in (0,1)$, zu approximieren. Das Bisektionsverfahren soll stoppen, wenn das halbierte Intervall in einem Schritt des Verfahrens eine Länge von 10^{-15} unterschreitet. Benutzen Sie die selbstgeschriebene Sinusfunktion auf $[-2,2]$ aus Aufgabe 7.2 b). Mit der Identität

$$\sin^2\left(\frac{\pi}{4}\right) = \frac{1}{2}$$

approximiert dieses Programm $\pi/4$.

(b) Benutzen Sie Ihre Approximation von $\pi/4$ aus b), die Periodizität der Sinusfunktion und die selbstgeschriebene Sinusfunktion auf $[-2,2]$ aus Aufgabe 7.2 b) um eine „globale“ Sinusfunktion zu implementieren. Vergleichen Sie die Ausgabe mit der vorimplementierten Sinusfunktion.

(c) Ergänzen Sie das vorhergehende Programm um `gnuplot`-Visualisierungen auf vom Benutzer einzugebenden Bereichen. Geben Sie bitte einen Ausdruck für das Intervall $[0,12]$ ab.

(3) Sei $\gamma(t) = (a\cos(t), b\sin(t)), t \in [0, 2\pi], a \geq b > 0$, eine Paramaterdarstellung einer Ellipse.

(a) Zeigen Sie mit Hilfe des Kurvenintegrals, dass für den Umfang $U = \int_0^{2\pi} \|\gamma'(t)\|\, dt$ einer Ellipse

$$U = 4a \int_0^{\pi/2} \sqrt{1 - \varepsilon^2 \sin^2(t)} dt$$

gilt, wobei $\varepsilon^2 = \frac{a^2 - b^2}{a^2}$.

(b) Beweisen Sie folgende Formel

$$U = 4a\frac{\pi}{2}\left(1 + \sum_{k=1}^{\infty}\left((-1)^k \varepsilon^{2k} \prod_{j=1}^{k} \frac{(2j-1)(3-2j)}{(2j)^2}\right)\right).$$

Hinweis. Beweisen Sie per vollständiger Induktion für $k \in \mathbb{N}$

$$\int_0^{\pi/2} \sin^{2k}(t) dt = \frac{\pi}{2} \prod_{j=1}^{k} \frac{2j-1}{2j}.$$

Sie dürfen ohne Beweis benutzen, dass die Binomische Reihe

$$(1+x)^{1/2} = \sum_{k=0}^{\infty} \binom{\frac{1}{2}}{k} x^k,$$

wobei $\binom{\frac{1}{2}}{k} = \prod_{j=1}^{k} \frac{\frac{1}{2}+1-j}{j}$, für $|x| < 1$ absolut konvergiert.

(c) Sei $2b \geq a \geq b$, d.h. $\varepsilon^2 \leq \frac{3}{4}$. Sei $\varepsilon \mapsto u(\varepsilon) = 1 + \sum_{k=1}^{\infty}\left((-1)^k \varepsilon^{2k} \prod_{j=1}^{k} \frac{(2j-1)(3-2j)}{(2j)^2}\right)$ und $T_K(\varepsilon) := 1 + \sum_{k=1}^{K}\left((-1)^k \varepsilon^{2k} \prod_{j=1}^{k} \frac{(2j-1)(3-2j)}{(2j)^2}\right)$. Bestimmen Sie K so, dass $\sup_{\varepsilon^2 \in [0,3/4]} |T_K(\varepsilon) - u(\varepsilon)| \leq 10^{-15}$.

(d) Implementieren Sie mit diesem K diese Approximation T_K in einem C-Programm. Die Werte a, b sollen vom Nutzer eingegeben werden. Benutzen Sie für $\pi/2$ den Wert `1.570796326794896`. Mit Aufgabe 2 können wir $\pi/2$ auch approximieren.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben.

Abgabe der Lösungen: Mittwoch, den 26.06.2013 bis 11.00 Uhr vor Raum 02-06 (Büro L. Pulst).

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Bitte beachten Sie die folgenden Informationen zur Klausur:

- *Die Klausur findet am Freitag, den 05.07.2013 in Raum G03-315 von 15.15 Uhr - 16.45 Uhr (d.h. Dauer 90 Minuten) statt. Bitte seien Sie pünktlich da.*
- *Bringen Sie bitte ausreichend Papier, Schreibzeug (Kugelschreiber oder Füller) und einen Lichtbildausweis mit.*
- *Während der Klausur dürfen Sie das* **gedruckte** *Vorlesungsskript von Prof. Grunau und ein festgebundenes Programmierhandbuch ihrer Wahl benutzen, allerdings keine losen Blätter-Sammlungen.*
- *Es sind keinerlei elektronische Hilfsmittel (Smartphones, Laptops, Netbooks, etc.) erlaubt.*

Prof. H.-Chr. Grunau
Dipl.-Math. L. Pulst

25.06.2013
Sommersemester 2013

ÜBUNGEN ZUR ALGORITHMISCHEN MATHEMATIK II
BLATT 10

(1) ★ Sei $B_1(0) \subset \mathbb{R}^2$, $u : \overline{B_1(0)} \to \mathbb{R}$ stetig und $u : B_1(0) \to \mathbb{R}$ zweimal stetig differenzierbar. Wir betrachten das folgende Randwertproblem:

$$\begin{cases} -\Delta u(x,y) = \lambda u(x,y) & \text{in } B_1(0), \\ u = 0 & \text{auf } \partial B_1(0), \end{cases} \tag{2}$$

$u \not\equiv 0$. Wir wollen den kleinsten Eigenwert λ bestimmen. Sei $J_0(r) = \sum_{k=0}^{\infty} \frac{(-1)^k}{4^k (k!)^2} r^{2k}$ die Besselfunktion aus den Aufgaben 7.3 und 8.1.

(a) Zeigen Sie, dass die Funktion $J_0(\sqrt{\lambda} r)$ die Differentialgleichung

$$\tilde{u}''(r) + \frac{1}{r}\tilde{u}'(r) + \lambda \tilde{u}(r) = 0$$

löst.

(b) Damit $J_0(\sqrt{\lambda} r)$ eine radialsymmetrische Lösung von (2) ist, muss $J_0(\sqrt{\lambda} \cdot 1) = J_0(\sqrt{\lambda}) = 0$ gelten. Beweisen Sie, dass $J_0(r)$ in $[0,3]$ eine Nullstelle besitzt.

Hinweis. Zeigen Sie, dass $\sum_{k=0}^{4} \frac{(-1)^k}{4^k (k!)^2} 9^k < 0$ und für alle $k = 5, 7, 9, \ldots$ gilt:

$$\frac{(-1)^k}{4^k (k!)^2} 9^k + \frac{(-1)^{k+1}}{4^{k+1}((k+1)!)^2} 9^{k+1} < 0.$$

(c) Implementieren Sie in einem C-Programm ein Bisektionsverfahren um die Nullstelle in $[0,3]$, und somit λ, von J_0 zu approximieren.

(2) (a) Sei $G \subset \mathbb{R}^n$ offen und konvex, $f : G \to \mathbb{R}$. f heißt (strikt) konvex, falls für je zwei Punkte $x_0, x_1 \in G$, $x_0 \neq x_1$ und für $t \in (0,1)$ gilt:

$$f\left(t x_0 + (1-t) x_1\right) \ (<) \ \leq t f\left(x_0\right) + (1-t) f\left(x_1\right).$$

Sei nun f zweimal stetig differenzierbar. Zeigen Sie, dass f genau dann konvex ist, wenn $\operatorname{Hess} f(x)$ für alle $x \in G$ positiv semidefinit ist. Ist $\operatorname{Hess} f(x)$ für alle $x \in G$ positiv definit, so ist f strikt konvex.

Hinweis. Eine symmetrische Matrix $A \in \mathbb{R}^{n \times n}$ heißt positv semidefinit, falls für alle $\xi \in \mathbb{R}^n$ gilt: $\xi^T \circ A \circ \xi \geq 0$. A heißt positiv definit, falls es ein $\lambda > 0$ gibt, so dass gilt: $\forall \xi \in \mathbb{R}^n : \ \xi^T \circ A \circ \xi \geq \lambda \|\xi\|^2$.

(b) Sei $f : \mathbb{R}^n \to \mathbb{R}$ zweimal stetig differenzierbar und $\operatorname{Hess} f(x)$ für alle $x \in \mathbb{R}^n$ positiv definit, $x_0 \in \mathbb{R}^n$, $R > 0$. Für alle $x \in \partial B_R(x_0)$ gelte

$$\langle \nabla f(x), x - x_0 \rangle > 0.$$

Zeigen Sie, dass f ein globales Minimum besitzt. Dieses liegt in $B_R(x_0)$.

(3) ★ Um lokale Minima von reellwertigen differenzierbaren Funktionen $f : \mathbb{R}^n \to \mathbb{R}$ zu suchen, kann man ein Gradientenverfahren benutzen. Sei dazu $x_k \in \mathbb{R}^n$ eine Näherung einer Minimalstelle. Mit der Schrittweite $\varepsilon > 0$ setzen wir

$$x_{k+1} = x_k - \varepsilon \nabla f(x_k).$$

Implementieren Sie in C-Progammen das Gradientenverfahren für die folgenden Funktionen. Der Benutzer soll die Möglichkeit erhalten den Startwert x_0, die Schrittweite ε und das Abbruchkriterium $\|x_{k+1} - x_k\| < \delta$ festzulegen. Schreiben Sie zwei Programme: in einem ersten Programm können Sie die analytisch ausgerechnete Ableitung benutzen. Für das zweite Programm soll der Differenzenquotient

$$\frac{f(x_0 + h) - f(x_0)}{h}$$

die Ableitung $f'(x_0)$ approximieren. Der Parameter h soll vom Benutzer eingegeben werden können.

(a) $f_1(x) = \dfrac{x^6 + x^5 + x^4 + x^3 + x^2 + x + 1}{x^4 + 1}$;
(b) $f_2(x) = 2\sin(2x - 5.8) - 2x + \exp(x - 2) + 6$.

(4) Implementieren Sie das Gradientenverfahren aus Aufgabe 2 für die Funktion

$$f(x, y) = \sin(x) + \sin(y) - \sin(x - y).$$

Visualisieren Sie das Verfahren in einem zweiten Programm, indem Sie sich die Schritte (x_k, y_k) von gnuplot darstellen lassen. Speichern Sie dazu die Werte x_k, y_k jeweils in dynamischen Arrays. Es reicht wenn Sie in diesem Programm den analytisch bestimmten Gradienten benutzen.

(5) Implementieren Sie in einem C-Programm mit Hilfe des Newtonverfahrens die Logarithmusfunktion. Obwohl es sollte, konvergiert dieses Verfahren für große $x \in (0, \infty)$ nicht. Dividieren Sie x solange durch e bis ein geeigneter Startwert erreicht ist. Visualisieren Sie mit Hilfe von gnuplot in einem zweiten Programm diese selbsgeschriebene Logarithmusfunktion auf vom Benutzer festgelegten Bereichen. Vergleichen Sie auch hier mit der vorimplementierten Logarithmusfunktion.

Die mit ★ gekennzeichneten Aufgaben sind Hausaufgaben.

Abgabe der Lösungen: Mittwoch, den 03.07.2013 bis 11.00 Uhr vor Raum 02-06 (Büro L. Pulst).

Bitte beachten Sie die Hinweise zur Abgabe von Programmieraufgaben. Bei Ihrer handschriftlichen Abgabe können Sie – analog zur Vorlesung – Standardinitialisierungs-, Ein- und Ausgabefunktionen summarisch darstellen.

Bitte beachten Sie die folgenden Informationen zur Klausur:

- *Die Klausur findet am Freitag, den 05.07.2013 in Raum G03-315 von 15.15 Uhr - 16.45 Uhr (d.h. Dauer 90 Minuten) statt. Bitte seien Sie pünktlich da.*
- *Bringen Sie bitte ausreichend Papier, Schreibzeug (Kugelschreiber oder Füller) und einen Lichtbildausweis mit.*
- *Während der Klausur dürfen Sie das* **gedruckte** *Vorlesungsskript von Prof. Grunau und ein festgebundenes Programmierhandbuch ihrer Wahl benutzen, allerdings keine losen Blätter-Sammlungen.*
- *Es sind keinerlei elektronische Hilfsmittel (Smartphones, Laptops, Netbooks, etc.) erlaubt.*