Grundlagen der Informatik

Entwurf von typischen Algorithmen zur Datenverarbeitung Teil 1

Prof. Dr.-Ing. Thomas Wiedemann
Fachgebiet Informatik / Mathematik

Überblick zur 4. Vorlesung

- Demonstration der Vorgehensweise zur Entwicklung von Algorithmen
- **Teil 1: Einfache mathematische und statistische Berechnungen**

Nach der Einführung von Feldern und Strukturen:

- Teil 2: Typische Algorithmen zur Bearbeitung ein- und mehrdimensionaler Felder (Sortieren und Suchen)
- Teil 3: Algorithmen zur Verwaltung dynamische Datenstrukturen (Bäume und Netze)

Wiederholung

Allgemein :

- Eine Aufgabe ist meist mit verschiedenen Algorithmen lösbar ! Zur qualitativen Bewertung sollten Effizienzmaße und Fragen der Anpassungs- und Wartungsfähigkeit herangezogen werden.

Aufgabe

⇩ **Denken !!!!!**

Algorithmus

- Gute Algorithmen setzen immer auch eine sehr gute Kenntnis des **jeweiligen Anwendungsgebietes** und zugehöriger **mathematischer Verfahren** voraus.

Hilfestellungen und sinnvolle Lösungsansätze

- Praktische Aufgaben lassen sich häufig durch eine Kombination verschiedener Basis- oder Teilalgorithmen lösen (siehe folgende Beispiele)
- Diese Basisalgorithmen sollen nachfolgend vorgestellt werden.
- Offene Probleme lassen sich meist durch Anpassung nur sehr kleiner Algorithmenabschnitte lösen. Dazu ist ein grundlegendes Verständnis der Abläufe und eine kreative Anpassung an konkrete Aufgaben notwendig.

1.1. Berechnung einfacher mathematischer Ausdrücke

1.1 Einfache mathematische Ausdrücke

$$y = f(x) = x^2 * \cos(3x) / (x-3)$$

- Die Umsetzung als Algorithmus kann in großer Nähe zur mathematischen Formulierung erfolgen !
- Zu beachten und gesondert zu definieren sind mögliche Probleme bei der konkreten numerischen Berechung der Ausdrücke, wie
  - Division durch 0 oder Werte nahe 0
  - Wurzel aus negativen Werten
  - Rechnung mit sehr großen Werten (Gefahr des Überlaufs)

Im obigen Fall wäre eine Division für X nahe bei 3 zu beachten :

1. eps = 10 E-7
2. Wenn abs(x-3) < eps dann Ausgabe ("Division durch 0 -> Y -> ∞") - Ende
3. y = x * x * cos(3 *x) / (x-3)
4. Ausgabe Y

1.2. Grundlegende Funktionsbausteine : Zählen

Setze Anfangswert : a = Startwert

1.2.1 Zählen aufwärts

a = a + 1 einfach Hochzählen

a = a + 2 in Zweierschritten Hochzählen

a = a + 10 in Zehnerschritten Hochzählen

Prüfung auf Ende: Wiederhole Zählaktion, falls a < Endwert

1.2.2 Zählen abwärts

a = a - 1 einfach Runterzählen

a = a - 2 in Zweierschritten Runterzählen

a = a - 10 in Zehnerschritten Runterzählen, ...

Prüfung auf Ende: Wiederhole Zählaktion, falls a > Endwert

1.2. Grundlegende Funktionsbausteine : Gruppieren / Bewerten

Aufgabe: Ordne Werte zu Gruppen (Gewicht -> Porto, Alter -> Gruppe)

1.2.3 Verschachtelte Bedingungen (Bsp. Wiesoft-Logistik)

Wenn G > 10 Kg dann Wenn G < 20 Kg dann P = 15 €
sonst Nicht Beförderbar
sonst Wenn G > 1kg dann P = G * 1,43€
sonst P = 3,40 € (Päckchen)

1.2.4 Bedingungen mit Überschreiben vorheriger Werte

Wenn G < 1kg dann P = 3,40 € (Päckchen)

Wenn G >= 1kg dann P = G * 1,43€ (Paket)

Wenn G > 10kg dann P = 15 € (Sperrgut)

Wenn G > 20kg dann P = -1 € (Kennung für nicht ZULÄSSIG)

Berechnung von Funktionsverläufen

1.3. Berechnung von Wertentwicklungen

- In allen Bereich von Technik und Wirtschaft verändern sich Werte in der Zeit :
  - Bewegung: Weg, Geschwindigkeit, Beschleunigung
  - Wärmelehre : Temperatur, Druck
  - Elektrotechnik / Elektronik: Spannung, Strom, Induktion, ...
  - Wirtschaft : Zinsen, Börsenkurse, Rentneranteil , ...
- Bei Zusammentreffen vieler Größen sind die bekannten Formeln aus der Theorie häufig kaum noch anwendbar, da nicht mehr zutreffend (bei einem Raketenstart gelten zwar alle Basisgesetze, doch durch den Treibstoffabbrand verändert sich fortlaufend das Gewicht der Rakete und damit die Beschleunigung !)
- Als Lösung ergibt sich damit nur die Option, in Näherung der Werteverlauf über die Zeit numerisch mit dem Rechner zu berechnen.
- Vertieft wird dieses Thema im Bereich Modellierung & Simulation in höheren Semestern !
- Exemplarisch soll die Vorgehensweise am Beispiel von Bewegungen gezeigt werden. Alle anderen Aufgabengebiete sind analog lösbar !

Berechnung von Funktionsverläufen II

1.3.1 Allg.Vorgehensweise (= Algorithmus für Funktionsdarst.)

Bei zeitlichen Entwicklungen ist zuerst der notwendige Takt der Zeit **Δt** zu bestimmen:

- Ingenieurgrundsatz: Nur so genau wie nötig, nicht so genau wie möglich!
- Rakete : **Δt = 0.1s ... 0.001s**
- PKW : **Δt = 3s ... 0.1s**
- Sinnvoll ist eine Prüfung der Taktzeit mit unterschiedlichen Werten ! Weichen die Ergebnisse stark ab (>5% Abweichung) so sind ggf. kleinere Taktzeiten oder bessere Berechnungsvorschriften zu verwenden (siehe Ma:Differentialgleichungen)
- Bei grafischen Darstellungen ist eine “saubere“, durchgezogene Kurvendarstellung zu erreichen ! (In Klausuren wird ggf. der Taktwert vorgegeben!)
- Weiterhin sind für das zu berechende System die Anfangswerte und die Endbedingungen zu bestimmen (Endlichkeit von Algorithmen):

- Rakete : Höhe0=0, V0=0, Treibstoff= 10t   Ende: Höhe wieder auf 0, Treibs=0
- Bremsprobe PKW: V0=100 Km/h, Beschleunig_Bremse= -2 m/s²   Ende: v=0

Berechnung von Funktionsverläufen III

1.3.2. Beispielalgorithmus einer gleichmäßigen Beschleunigung

Festlegen aller Anfangswerte (die Einheiten können später nicht geschrieben werden):

- **T=0**
- **Delta_t = 0.1 [s]**
- **Weg = 0**
- **V0=0 [m/s]**
- **A= 2 [m/s²]**

Durchführen EINES Zeitschrittes (von einigen 10.000) mit bekannten Formeln

6. **T = T + Delta_T**
7. **V = A * T**
8. **S = ½ * A * T²**
9. **Ausgabe T, V, S**

Prüfung auf Ende erreicht ?

10. **Wenn V<100/3.6 dann Gehe zu 6. (mache weiter)**
11. **Ausgabe „Zeit für 100km/h:“ T**
12. **Ende**

Berechnung von Funktionsverläufen IV

1.3.3. Beispielalgorithmus einer ungleichmäßigen Beschleunigung

- **T=0**
- **Delta_t = 0.1 [s]**
- **S = 0**
- **V=100 [m/s]**
- **A0= -2 [m/s²]**

Durchführen EINES Zeitschrittes – durch A<>const gelten die bekannten Formeln nicht MEHR, daher jetzt näherungsweise Berechnung mit kleinen Inkrementen

6. **A = A0 * (1 - (V / 120)²) (Aquaplaning !!!! Formel korrekt ???)**
7. **V = V + A * Delta_T**
8. **S = S + V * Delta_T**
9. **T= T + Delta_T**
10. **Ausgabe T, V, S**

Prüfung auf Ende erreicht ?

11. **Wenn V>0 dann Gehe zu 6. (mache weiter)**
12. **Ausgabe „Bremsweg =“ S , ENDE**

Berechnung von Summen und Produkten

2a. Endliche Summen und Produkte

- viele Berechnungen erfordern die Bestimmung von Summen und Produkten mit einer endlichen (=n) Anzahl von Gliedern

z.B. Mittelwert = $\overline{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$

oder allgemeiner $y = f_2(x) \sum_{i=a}^{n} f_1(x,i)$

$$y = f_2(x) \prod_{i=a}^{n} f_1(x,i)$$

Summenberechnung – Algorithmische Umsetzung

Allgemeine Vorgehensweise am Beispiel der Summenberechnung :

$$y = f_2\left(x,a,n, \sum_{i=a}^{n} f_1(x,i)\right)$$

1. Die **Summe wird als ein Zyklus** vom Startwert a bis zum Endwert n umgesetzt !
2. Innerhalb des Zyklus werden die einzelnen **Glieder $f_1(x,i)$ auf die Variable Summe aufaddiert**.
3. Die Variable Summe hat in der Regel den Anfangswert 0.
4. Nach Beendigung des Zyklus ist die abschließende Funktion f_2 zu berechnen.

Summenberechnung

Summe = 0

i = a (1) n

Summe = Summe $+f_1(x,i)$

Summe = f$_2$(x,a,n,Summe)

Produktberechnung – Algorithmische Umsetzung

Zur Produktberechnung sind nur geringe Änderungen notwendig :

$$y = f_2\left(x,a,n, \prod_{i=a}^{n} f_1(x,i)\right)$$

1. Das **<u>Produkt wird als ein Zyklus</u>** vom Startwert a bis zum Endwert n umgesetzt !
2. Innerhalb des Zyklus werden die einzelnen **<u>Glieder $f_1(x,i)$ auf die Variable Produkt</u>** **<u>aufmultipliziert</u>**.
3. Die Variable Produkt hat in der Regel den **<u>Anfangswert 1</u>**.
4. Nach Beendigung des Zyklus ist die abschließende Funktion f_2 zu berechnen.

Produktberechnung

- **Produkt = 1**
- **i = a (1) n**
  - **Produkt = Produkt** * $f_1(x,i)$
- **Produkt =** f_2(x,a,n,**Produkt**)

Summenberechnung – Beispiel

Demonstration am Beispiel statistische Standardabweichung :

$$\sigma = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

1. Die **<u>Summe wird als ein Zyklus</u>** vom Startwert 1 bis zum Endwert n umgesetzt !
2. Innerhalb des Zyklus werden die einzelnen **<u>Glieder</u>** $(x_i - x)^2$ **<u>auf die Variable Summe</u>** **<u>aufaddiert</u>**.
3. Die Variable Summe hat den Anfangswert 0.
4. Nach Beendigung des Zyklus ist die abschließende Division durch (n-1) zu berechnen.

Summenberechnung

- **Summe = 0**
- **i = 1 (1) n**
  - **Summe = Summe** $+(x_i - x)^2$
- **Summe** = Summe / (n-1)

Sonderfälle ? Probleme ?

Berechnung von Summen und Produkten

2b. Unendliche Summen

- Viele mathematischen Funktionen oder Konstanten können numerisch nur durch Näherungsformeln auf der Basis von Summen berechnet werden.
- Beispiel: Die Eulersche Zahl ist berechenbar durch

$$e = \sum_{i=0}^{\infty} \frac{1}{i!}$$

Problem

- eine endlose Reihe ist jedoch nicht berechenbar !
- Kann an geeigneter Stelle abgebrochen werden ?
- **Ja: Wenn der Partialsummenausdruck keine Änderung mehr bewirkt !**

Wann ist dies der Fall ?

- Wenn das Verhältnis von Summenwert und Partialsumme in die Nähe der Rechengenauigkeit kommt !
- **Grundvoraussetzung : Konvergenz der Partialsumme gegen 0**

Berechnung der Eulerschen Zahl – Algorithmische Umsetzung

Genau nach dieser allgemeinen Vorgehensweise läßt sich e berechnen :

$$y = \sum_{i=1}^{\infty} \frac{1}{i!}$$

1. Die **Summe wird wieder als ein Zyklus** beginnend vom Startwert 1 umgesetzt
2. Innerhalb des Zyklus werden die einzelnen **Glieder $f_1(x,i)$ auf die Variable Summe aufaddiert**. (Summe mit Anfangswert 0)
3. Als **Abbruchbedingung wird ein Unterschreiten einer Schranke eps** durch die Partialsumme verwendet !
4. Nach Beendigung des Zyklus ist die abschließende Funktion f_2 zu berechnen.

```
Summenberechnung
  eps = 10 E-8
  Summe = 0     i = 1
    PSumme = 1 / ( fak(i) )
    Summe = Summe +PSumme
    i = i+1
  Solange  abs(PSumme) > eps
  Ausgabe e = Summe
```

Unendliche Summenberechnung – Algorithmische Umsetzung

Allgemeine Vorgehensweise am Beispiel der Summenberechnung :

$$y = f_2\left(x,a,n, \sum_{i=1}^{\infty} f_1(x,i)\right)$$

1. Die **Summe wird wieder als ein Zyklus** beginnend vom Startwert a umgesetzt
2. Innerhalb des Zyklus werden die einzelnen **Glieder $f_1(x,i)$ auf die Variable Summe aufaddiert**. (Summe mit Anfangswert 0)
3. Als **Abbruchbedingung wird ein Unterschreiten einer Schranke eps** durch die Partialsumme verwendet !
4. Nach Beendigung des Zyklus ist die abschließende Funktion f_2 zu berechnen.

Summenberechnung

eps = 10 E-8

Summe = 0 i = a

PSumme = f_1(x,i)

Summe = Summe +PSumme

i = i+1

Solange abs(PSumme) > eps

Summe = f_2(x,a,n,Summe)

Effizienzbetrachtungen bei der Berechnung unendlicher Summen

- Als Funktionen zur Berechnung von Partialsummen tauchen sehr häufig Fakultäten oder Potenzen auf.
- Beispiel: **Berechnung des cosinus**

$$y = \cos(x) = 1 + \sum_{i=1}^{\infty} (-1)^{i} \frac{x^{2i}}{(2i)!} \qquad \text{für } (0 < X < Pi/2)$$

- Durch den n-fachen Aufruf von Fakultätsfunktionen oder die Berechnung von Potenzen (welche auch wiederum als Mehrfachprodukt oder Reihenentwicklung entwickelt werden müssen) kommt es zu einem quadratischen Anwachsen der Rechenzeiten, die Effizienz ist vom Typ $O(n^2)$!
- Es stellt sich die Frage, ob die Effizienz verbessert werden kann ?
- Bei einer manuellen Entwicklung der Reihen ist auffällig, daß bei Formeln obigen Typs die i+1. Partialsumme alle Rechenschritte bis zum i. Glied mit dem Vorgänger gemeinsam hat.

Verbesserungen der Effizienz bei unendlichen Summen

Allgemeine Vorgehensweise am Beispiel der cos-Berechnung :

$$y = \cos(x) = \sum_{i=1}^{\infty} (-1)^{i} \frac{x^{2i}}{(2i)!} \quad \text{für } (0 < X < Pi/2)$$

1. Der **Partialsummenausdruck** ist in einzelne Glieder zu zerlegen, welche getrennt berechenbar sind :

   Vorzeichen = $(-1)^{i}$

   Potenz = x^{2i}

   Fakultaet = $(2i)$

   Psumme = Vorzeichen * Potenz / Fakultaet

2. Es sind die Beziehungen zwischen dem i. und i-1. Glied zu bestimmen :

   $\text{Vorzeichen}_{i} = \text{Vorzeichen}_{i-1} * (-1)$

   $\text{Potenz}_{i} = \text{Potenz}_{i-1} * x * x$

   $\text{Fakultaet}_{i} = \text{Fakultaet}_{i-1} * (2i+1) * (2i+2)$

3. Es sind abhängig von 2. korrekte Startwerte für die Einzelausdrücke festzulegen:

   Vorzeichen = 1 (da $\text{Vorzeichen}_{1} = -1$) Potenz =1 Fakultaet = 1