Inhalt

1. Einführung in die Informatik

2. Algorithmen

- **Definition, Eigenschaften, Entwurf**
- Darstellung von Algorithmen
- Beispiele

2. Algorithmen

Einführende Beispiele

Täglich werden Verarbeitungsvorschriften ausgeführt, die einen Weg beschreiben, wie ein gewünschtes Ergebnis erzielt werden kann. Solche Vorschriften sind:

- **Bedienungsanleitungen**

  zum Beispiel zur Bedienung eines Parkscheinautomaten
- **Montageanleitungen**

  für Möbel, für Modellbaukästen, usw.
- **Verhaltensvorschriften**

  Verhalten bei Unfällen, Feueralarm, usw.
- **Rezepte**

  Mischen von Farben, Kochrezepte, usw.
- **Rechenvorschriften**

  Lösen eines Gleichungssystems, Ermittlung statistischer Kenngrößen, Stöchiometrie (Chemisches Rechnen), Bestimmung von Flächeninhalten, usw.

Algorithmen

Algorithmen sind spezielle Verhaltensvorschriften, die gewisse Mindestanforderungen erfüllen müssen.

Algorithmendefinition

In diesem Abschnitt soll der Algorithmenbegriff etwas genauer präzisiert werden. In der Literatur gibt es dazu verschiedene Definitionen.

Drei solcher Definitionen sollen hier kurz angegeben werden:

Algorithmen

1. Ein Algorithmus ist eine eindeutige Vorschrift zur Lösung einer Aufgabe. Die Vorschrift besteht aus einer Folge von Anweisungen. Die Anweisungen beziehen sich auf Operationen, die jeweils den nächsten Lösungsschritt veranlassen. Die Anzahl der auszuführenden Lösungsschritte ist endlich und ihre zeitliche Reihenfolge liegt fest.

2. Ein Algorithmus ist eine präzise, d.h. in einer festgelegten Sprache abgefasste, endliche Beschreibung eines allgemeinen Verfahrens unter Verwendung ausführbarer elementarer Verarbeitungsschritte.

3. Ein Algorithmus ist eine detaillierte und explizite Vorschrift zur schrittweisen Lösung eines Problems. Im einzelnen bedeutet dies:
   - Die Ausführung erfolgt in diskreten Schritten
   - Jeder Schritt besteht aus einer einfachen, offensichtlichen Grundaktion
   - Zu jedem Zeitpunkt muss eindeutig bestimmt sein, welche Schritte als nächste auszuführen sind.

Algorithmen

Definition:

Ein Algorithmus ist eine Vorschrift zur Lösung einer Klasse von Problemen. Er besteht aus einer endlichen Folge von Schritten, mit der aus bekannten Eingangsdaten neue Ausgangsdaten eindeutig berechnet werden können.

Eigenschaften:

- Die Definition zeigt den engen Zusammenhang mit **Funktionen** (Eingangsdaten → Ausgangsdaten).
- Die Definition bindet den Algorithmus an die Lösung einer Klasse von Problemen und nicht an eine einzelne Aufgabe.

Problem:

- Die Definition eines Algorithmus enthält keine Forderungen an die praktische Ausführbarkeit des Algorithmus auf einer realen Maschine (Rechner).

Algorithmen als mathematische Funktionen (1)

Ein Algorithmus kann als Funktion (mathematische Definition) aufgefasst werden.

$y=f(x)$ oder $f: D \rightarrow Z, x \rightarrow y$

(D: Definitionsmenge, Z: Zielmenge)

In der Mathematik ist eine **Funktion** oder **Abbildung** eine Beziehung zwischen zwei Mengen, die jedem Element der einen Menge (Funktionsargument, unabhängige Variable, x-Wert) genau ein Element der anderen Menge (Funktionswert, abhängige Variable, y-Wert) zuordnet.

(Quelle: de.wikipedia.org)

Interpretation:

- Elemente aus Mengen: Eingabe und Ausgabe
- Die Elemente können auch Mengen sein

Algorithmen als mathematische Funktionen (2)

Somit wird einer Funktion ein Algorithmus zugeordnet.

Es gibt aber Funktionen, denen kein Algorithmus zugeordnet werden kann. Diese Funktionen sind damit nicht berechenbar.

Definition:

Eine Funktion f heißt berechenbar, wenn es einen Algorithmus gibt, der für jedes Argument x den Funktionswert f(x) berechnet.

Beispiele folgen …

Beispiele berechenbarer und nicht berechenbarer Funktionen (1)

a) Berechne alle Primzahlen im Intervall [1,100] !
Dieses Beispiel stellt eine berechenbare Funktion dar, der Algorithmus ist relativ einfach.

b) Berechne die Funktionswerte der Funktion

$$f(x+1) = 2^{f(x)} \text{ mit } f(0)=0 \text{ für } 6<=x<=12 \text{ !}$$

Dies ist auch eine berechenbare Funktion, die sich aber auf keiner realen Rechenmaschine abarbeiten lässt, da f(6) bereits eine Zahl mit 19000 Dezimalstellen ist.

c) Drucke alle ganzen Zahlen von 1 bis 10^{100} !
Dieses Beispiel ist berechenbar, aber nicht in angemessener Zeit auf einem realen Rechner ausführbar!

Beispiele berechenbarer und nicht berechenbarer Funktionen (2)

Fortsetzung:

d) „Berechne“ alle reellen Zahlen im Intervall [0,1]
Dieses Beispiel ist eine nicht-berechenbare Funktion.
Es kann bewiesen werden, dass solch ein Algorithmus nicht existieren kann.

Schlussfolgerungen:

- Einige Funktionen sind prinzipiell nicht berechenbar.
- Manche Funktionen sind unter realistischem Ressourcen- und Zeitaufwand nicht auf einem Rechner umsetzbar, obwohl sie prinzipiell berechenbar sind.
- Es ist zweckmäßig, an Algorithmen solche Forderungen zu stellen, dass sie auf realen Maschinen praktisch ausgeführt werden können.
  Im folgenden werden diese wichtigen Eigenschaften aufgeführt.

Charakteristische Eigenschaften von Algorithmen

Endlichkeit

Ein Algorithmus muss aus einer endlichen Anzahl von Lösungsschritten bestehen und nach Abarbeitung dieser endlich vielen Schritte nach einer endlichen Zeit das Ende erreichen.

Eindeutigkeit

Die einzelnen Schritte eines Algorithmus und ihre Aufeinanderfolge müssen eindeutig beschrieben sein.

Allgemeinheit

Ein Algorithmus darf nicht nur die Lösung einer speziellen Aufgabe (z.B. Lösung der Gleichung $x^2 + 2x + 1=0$) , sondern muss die Lösung einer Klasse von Problemen (z.B. die Lösung aller quadratischen Gleichungen $ax^2 +bx +c =0$) beschreiben.

Charakteristische Eigenschaften von Algorithmen

Determiniertheit (für deterministische Algorithmen)

Die mehrmalige Anwendung des Algorithmus mit denselben Eingangsdaten muss immer wieder dieselben Ausgangsdaten liefern.

Hinweis: Es gibt auch nichtdeterministische Algorithmen

Effizienz

Ein Algorithmus muss möglichst wenig Ressourcen einer Maschine, d.h. möglichst wenig Rechenzeit und möglichst wenig Speicher in Anspruch nehmen.

In der Informatik wurden Effizienzmaße für Algorithmen, die sogenannte **Zeit- und Speicherkomplexität** entwickelt, mit der Algorithmen bewertet werden können. Dies ist praktisch sehr wichtig, da oft zur Berechnung einer Funktion mehrere Algorithmen angegeben werden können.

Die Darstellung von Algorithmen

Zur Darstellung von Algorithmen sind Grundelemente notwendig.

Erklärung am Beispiel der Lösung einer quadratischen Gleichung

$$ax^2 + bx + c = 0$$

nach der Vorschrift:

$$x_{1/2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Neben der Notation einzelner **elementarer Aktionen/Anweisungen**, wie z.B. „Gebe die Werte der Koeffizienten a,b,c ein“ sind auch **logische Elementarstrukturen** notwendig, die die **zeitliche Ablauffolge** darstellen, wie z.B. die Folge (**Sequenz**) von Anweisungen

1. Gebe die Werte der Koeffizienten a,b,c ein
2. Berechne d = b*b – 4*a*c

aber auch die Auswahl (**Selektion**)

3. Falls d>=0 setze fort mit Schritt …

Die Darstellung von Algorithmen

Vollständiger Algorithmus mit nummerierten Anweisungen:

1. Gebe Werte der Koeffizienten a,b,c ein
2. Berechne d = b*b-4*a*c
3. Falls d>=0 setze fort mit Schritt 6
4. Ausgabe: "Keine Lösung"
5. Stop
6. Berechne x1 = (-b+sqrt(d))/(2*a)
7. Berechne x2 = (-b-sqrt(d))/(2*a)
8. Falls d>0 setze fort mit Schritt 11
9. Ausgabe: "Eine Lösung: ",x1
10. Stop
11. Ausgabe: "Zwei Lösungen: ",x1, " und ", x2
12. Stop

Hier enthalten:

- Sequenz, d.h. Ausführung der Anweisungen nach nummerierter Reihenfolge
- Selektion, hier Überspringen (bzw. Aufführen) von Teilsequenzen

Charakteristische Eigenschaften von Algorithmen

Beispiel: Finden des minimalen und maximalen Werts in einer Datenreihe mit n Werten.

Algorithmus A:

Durchlaufen aller Werte, Merken und Korrektur des bisherigen minimalen, maximalen Werts:

O(n), d.h. Zeitaufwand hängt hauptsächlich von n ab.

Eine Verdopplung der Datenmenge bewirkt eine Verdopplung der Laufzeit.

Algorithmus B (keine gute Idee):

Sortieren und danach Zugriff auf erstes und letztes Element

$O(n \log_2 n)$, d.h. Aufwand ist generell höher (Faktor $\log_2 n$)

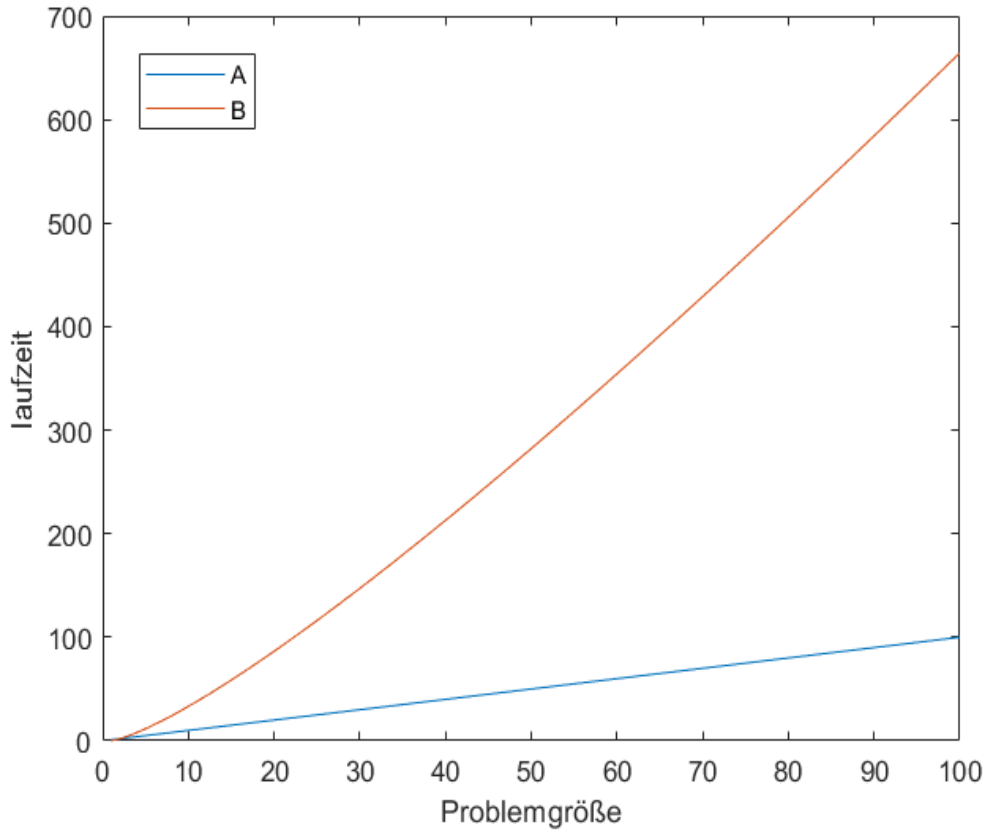
Ein Verdopplung der Datenmenge von n auf 2n führt zu einer Laufzeitverlängerung um Faktor $2 * (1+ \log_2 n)/ \log_2 n$

Charakteristische Eigenschaften von Algorithmen

Beispiel: Finden des minimalen und maximalen Werts in einer Datenreihe mit n Werten.

Algorithmus A:

Durchlaufen aller Werte, Merken und Korrektur des bisherigen minimalen, maximalen Werts:

Algorithmus B (keine gute Idee):

Sortieren und danach Zugriff auf erstes und letztes Element