

Algorithmen und Datenstrukturen

30.10.2007

1 Einführung: Algorithmen

1.1 Was ist ein Algorithmus?

Wikipedia:

„Unter einem Algorithmus versteht man allgemein eine genau definierte Handlungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen in endlich vielen Schritten.“

Beispiele:

- Kochrezept
- Bedienungsanleitung
- Notenblatt
- Programmablaufplan

Kein Computer ohne Algorithmen!

Keine Informatik ohne Algorithmen!

Kein Informatikstudium ohne Algorithmen!

Die Wikipedia-Definition gibt einem zunächst einmal eine einigermaßen anschauliche Vorstellung, gerade wenn man eine sehr genaue Vorstellung davon hat, was man für ein Problem lösen möchte und wie eine Lösung aussieht! Allerdings ~~ist~~ sind die Begriffe z.T. etwas vage...

Also: Was ist "ein Problem"?
 Was ist "eine Art von Problem"?
 Was ist "die Lösung eines Problems"?

Konkretes Beispiel:

- (1) Bestimme den größten gemeinsamen Teiler von 144 und 729! ← konkret
- (2) Bestimme den größten gemeinsamen Teiler zweier Zahlen x und y ! ← allgemein

Lösung:

- (1) 9
- (2) Euklidischer Algorithmus (siehe VL Programmierung!)

Also: Nicht verwirren lassen durch die verschiedenen Ebenen!
Dabei hilft saubere Schreibweise und Sprachgebrauch:

Problem! (Größter gemeinsamer Teiler)

Gegeben: Zwei positive ganze Zahlen, x und y

Gesucht: Die größte ganze Zahl z mit der Eigenschaft, dass z sowohl x als auch y teilt.

Eine derartig allgemein beschriebene ~~Aufgabenstellung~~ Aufgabenstellung bezeichnen wir als Problem.

Eine Aufgabenstellung mit konkreten Variablengrößen bezeichnen wir als Instanz eines Problems.

Mehr Beispiele im nächsten Kapitel!

Die Lösung eines Problems besteht nun in einem systematischen Verfahren, das einem für jede beliebige Instanz eine Lösung für diese Instanz liefert.

Also:

Gegeben: Problembeschreibung

Instanz { Gegeben: Inputzahlen
 { Gesucht: Outputzahlen } ↷

Gesucht: Systematisches Lösungsverfahren
 ↳ Algorithmus!

An so ein systematisches Lösungsverfahren stellt man noch einige Anforderungen:

(Eigenschaften nach Donald Knuth) * 10.01.1938

Ein Algorithmus besitzt die Eigenschaften

- **Endlichkeit** : Er endet immer nach einer endlichen Zahl von Schritten
- **Bestimmtheit** : Jeder Schritt eines Algorithmuses ist präzise definiert; die auszuführenden Handlungen sind streng und unmissverständlich für jede Situation beschrieben.
- **Funktionalität** : Ein Algorithmus hat keine oder mehrere Inputs, d.h. Größen, die ihm gegeben werden bevor er startet. Diese Inputs werden aus einer vorbestimmten Menge gewählt. Ein Algorithmus hat einen oder mehrere Outputs, d.h. Größen die in einem bestimmten ~~Verhältnis~~ Relation zum Input stehen.
- **Effektivität** : Alle auszuführenden Operationen müssen hinreichend elementar sein, dass sie im Prinzip präzise und in endlicher Zeit von einer Person mit Papier und Bleistift ausgeführt werden können.

Über einige dieser Eigenschaften kann man diskutieren; z.B. gibt es auch sog. randomisierte Algorithmen bei denen zu bestimmten Zeitpunkten „gewürfelt“ wird und das Würfelergebnis den nächsten Schritt bestimmt.

Nach ~~zwei~~ wichtiger Aspekt: Raum und Zeit!

(1) Ein Algorithmus darf in jedem Schritt nur endlich viel Speicherplatz in Anspruch nehmen

(2) Ein Algorithmus darf nur endliche Zeit in Anspruch nehmen.

~~Entscheidend~~
Das misst man mit dem Begriff der Komplexität:
Den mindestens benötigten Speicherplatz für einen Algorithmus nennt man seine Speicherkomplexität;
die mindestens benötigte Zahl von Schritten ist seine Zeitkomplexität

1.2 Historie

ca. 300 v. Chr.: Euklidischer Algorithmus
(ältester bekannter nichttrivialer Algorithmus!)

„Dixit Algorismus“

ca. 825 All-Chowarizmi:

Über das Rechnen mit indischen Zahlen

$$(XLVIII + XCII = CXL)$$

Römisch: $XLII + LXXIX = CXXI$

(römisch: „Er rechnet gut“ $\hat{=}$ „Er ist geizig“)

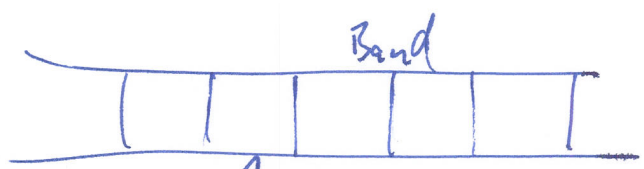
Indisch, d.h. dezimal:

$$42 + 79 = 121$$

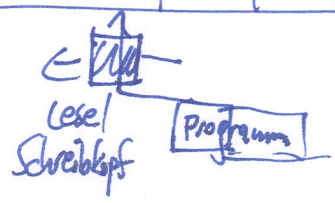
$$(42 + 92 = 140 \rightarrow \text{Turingbild!})$$

(\rightarrow Wichtigkeit von Datenstrukturen schon bei ganz elementaren Objekten!)

Turingmaschine: Abstraktion eines Algorithmus,
gleichzeitig aber sehr nah an realen Computern:



(1936)



Alan Turing (1912-1954)