Inhalt

1. Einführung in die Informatik

2. Algorithmen

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

Beispiele für Algorithmen

Im folgenden Abschnitt sollen ausgewählte Algorithmen, die spezielle Konstrukte des algorithmischen Paradigmas, wie

- Selektion, Mehrfachselektion
- Zyklen verschiedener Formen

enthalten, entwickelt und besprochen werden.

In einem weiteren Abschnitt werden dann Algorithmen mit Prozeduren und Funktionen behandelt und schließlich werden rekursive Algorithmen dargestellt. Ein Abschnitt zur Methodik des Algorithmenentwurfs schließt die Ausführungen über Algorithmen ab.

Algorithmen mit Selektion

Algorithmen, die nur die Konstrukte Sequenz und Selektion (einschließlich Mehrfachselektion) enthalten sind eigentlich sehr einfache Algorithmen, wenn man einmal ausschließt, dass Prozeduren aufgerufen werden. Die meisten anspruchsvollen Algorithmen enthalten Zyklen.

Beispiel 1: Berechnung Flächeninhalt eines Dreiecks

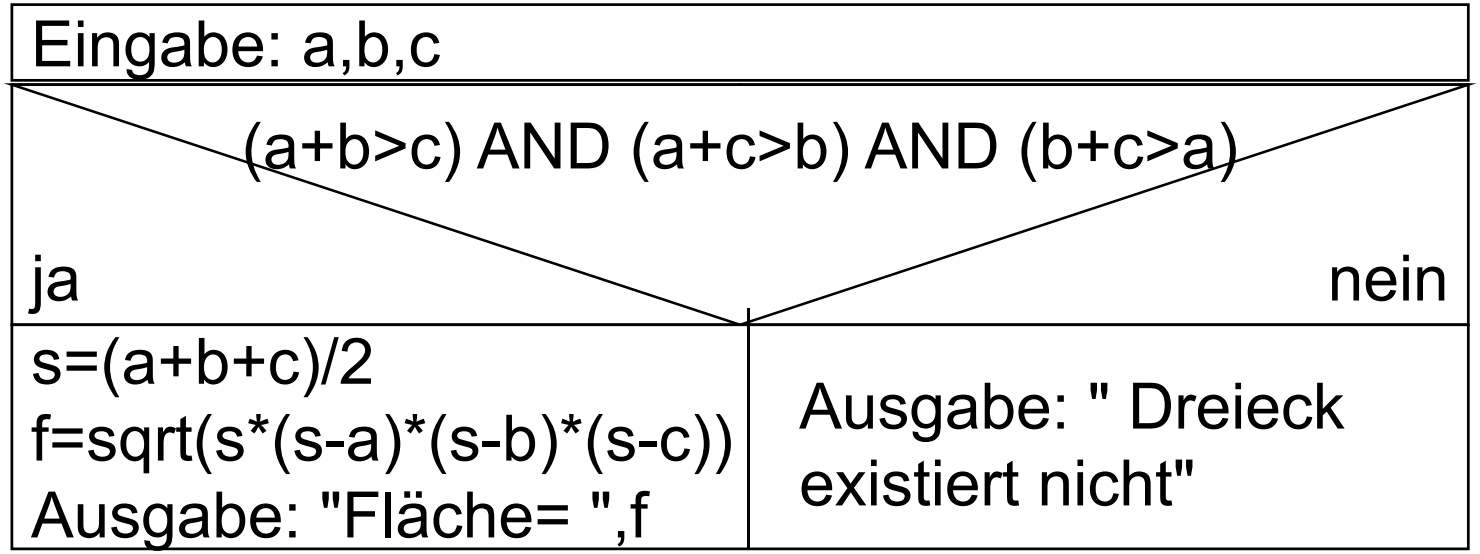
Aufgabe : Ein Dreieck mit den Seiten a,b,c hat einen Flächeninhalt

f= $\sqrt{s*(s-a)*(s-b)*(s-c)}$ mit s=(a+b+c)/2

Es ist zu prüfen, ob mit den Seiten wirklich ein Dreieck bildbar ist!

Algorithmen mit Selektion

Beispiel 1: Ein Dreieck ist nur bildbar, falls a+b>c und a+c>b und b+c>a gilt. Damit können wir folgenden Algorithmus angeben:



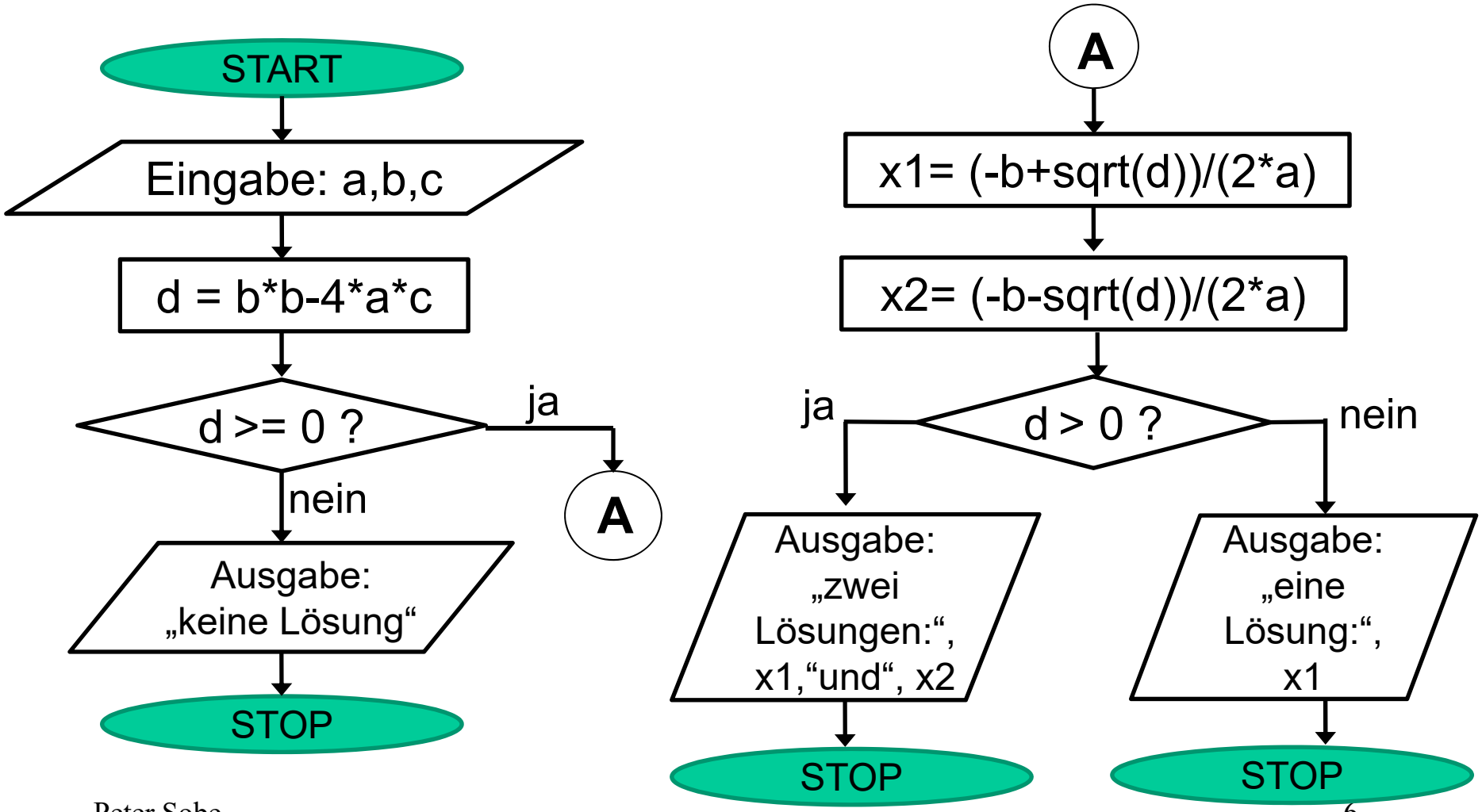
Algorithmen mit Selektion

Beispiel 2: Algorithmus, der ein gegebenes Datum, bestehend aus den drei Größen t für Tag, m für Monat und j für Jahr, überprüft, ob es sich um ein gültiges Datum aus dem Zeitraum 1583 bis 2100 handelt!

Die Lösungen werden zum Teil an der Tafel demonstriert, oder in der Übung erarbeitet.

Algorithmen mit Selektion

Lösung einer quadratischen Gleichung als PAP



Algorithmen mit Selektion und Zyklen

Zyklen in Algorithmen

- werden benutzt, um mehrere Datenelemente (Elemente innerhalb Mengen, Reihen, Feldern, Vektoren, Matrizen) nacheinander zu verarbeiten.

- sind nötig für schrittweise Lösungsverfahren, die sich durch die wiederholte Ausführung einer Regel einer Lösung annähern.

- werden für Verfahren verwendet, die kontinuierlich neue Daten einlesen und verarbeiten

Algorithmen mit Selektion und Zyklen

Beispiel 3: (Basisalgorithmus mit Zählschleife)
Es ist ein Algorithmus zur Bestimmung des Minimums einer Zahlenfolge a_1, a_2 ,... a_n aufzustellen.

Beispiel 4: (Basisalgorithmus mit geschachtelten Zählschleifen)
Es ist eine Weiterentwicklung des Algorithmus 4 vorzunehmen, der eine gegebene Zahlenfolge sortiert, indem er das Minimum bestimmt, an die 1.Position setzt und dann mit dem Rest (ab Position 2) wieder das Minimum bestimmt usw.

Sortieren durch Selektion und Zyklen (1)

Im folgenden soll ein sehr einfacher – intuitiver – **Sortieralgorithmus** angegeben werden.

Bubble Sort

Eingabe: feld a mit Elementen a(1) bis a(n), Feldlänge n
Ausgabe: feld a mit sortierten Elementen

Idee:

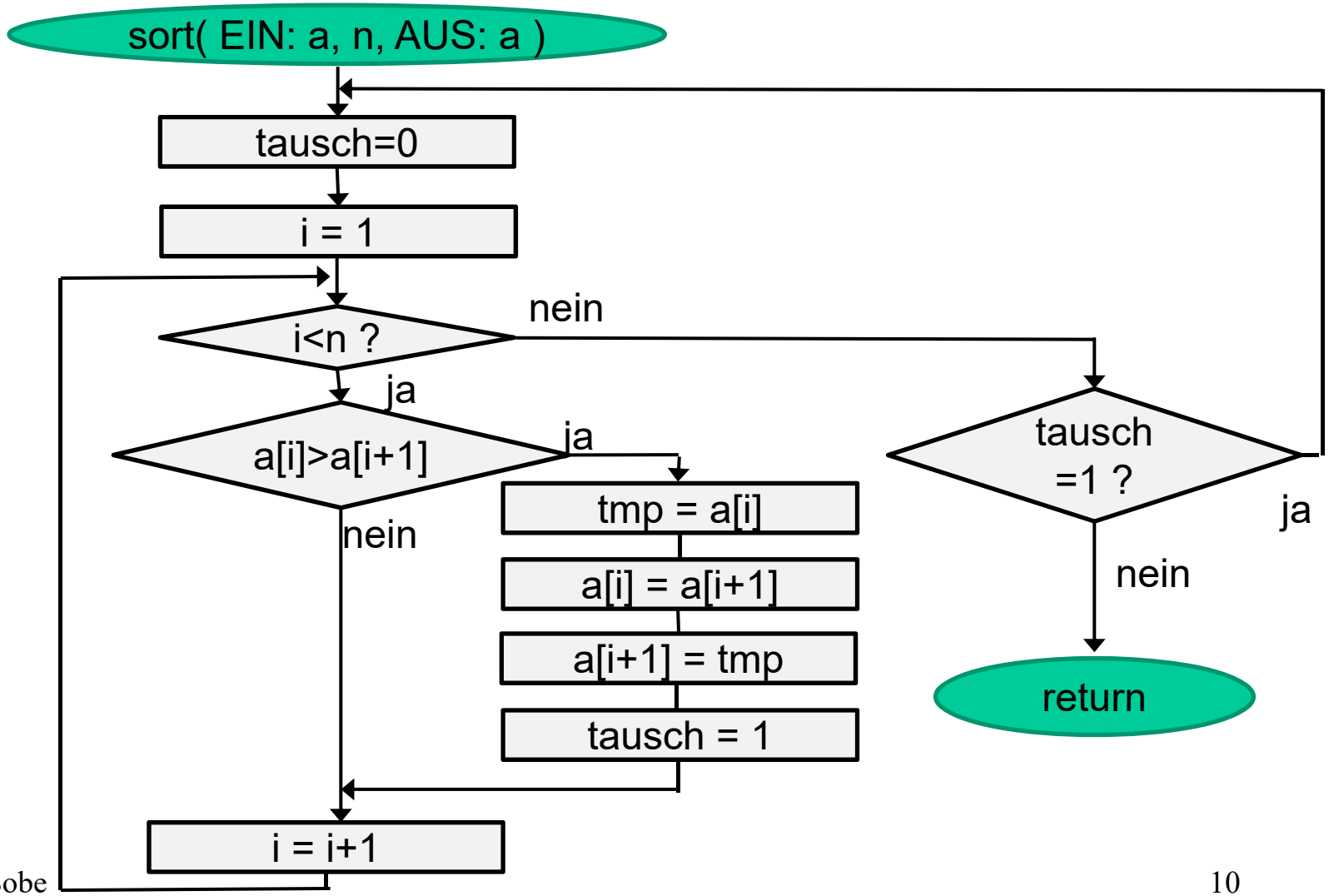
(1) Durchlaufe Feld mit aufsteigendem Index und vertausche benachbarte Felder, falls Sie nicht der geforderten Sortierreihenfolge entsprechen.

Wiederhole (1) solange, bis keine Vertauschungen mehr nötig sind

Sortieren durch Selektion und Zyklen (2)

Bubble Sort

PAP



Rekursive Algorithmen

In der Mathematik sind viele Funktionen rekursiv definiert.

Der Begriff der Rekursion beinhaltet, dass zur Definition einer Funktion diese selbst wieder mit benutzt wird, allerdings mit anderen Argumenten. Eine rekursive Definition benötigt stets eine (nichtrekursive) Anfangs- bzw. Abbruchbedingung.

Beispiel: Fakultät

rekursive Definition : fak(n) = n * fak(n-1)
Anfangsbedingung: fak(0) = 1

Rekursive Algorithmen - Charakteristika

Eine solche Definition ist in der Regel kurz und übersichtlich.

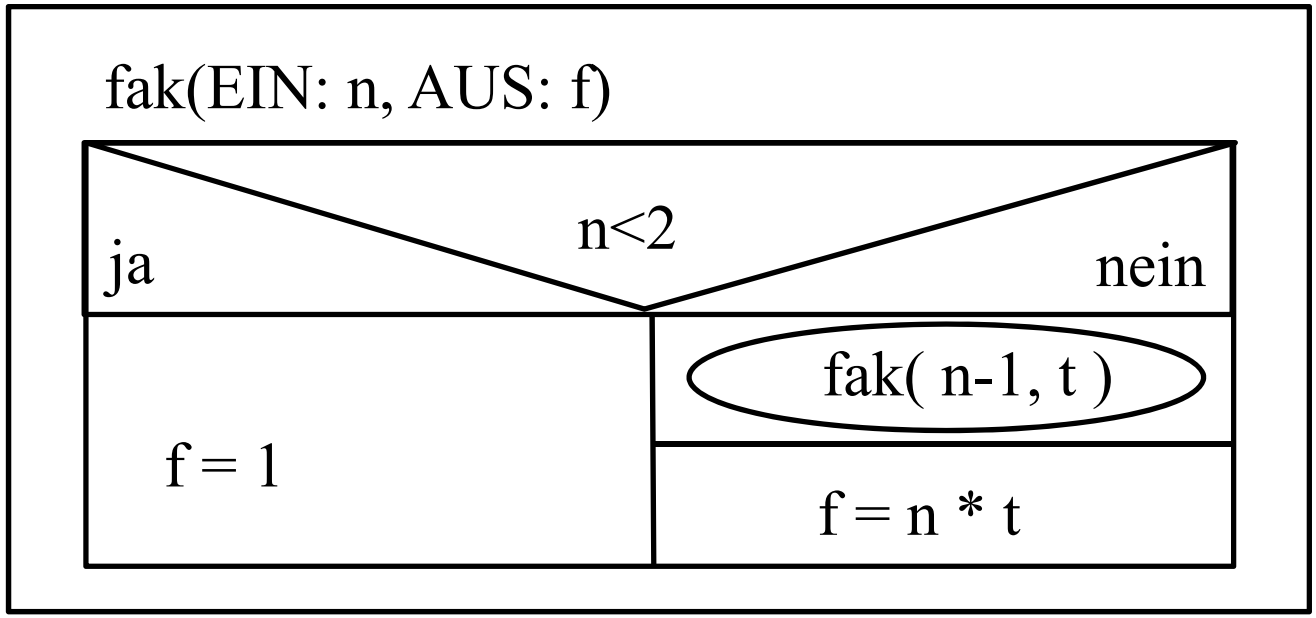
Man erkennt sofort die Grundstruktur des Algorithmus. Eine solche rekursive Definition lässt sich auch sehr leicht unter Verwendung rekursiver Prozeduren (Funktionen in C) implementieren. Allerdings ist der implementierte Algorithmus häufig ineffizient, d.h. mit einem hohen Ressourcenverbrauch (Speicher und Rechenzeit) verbunden.

Bestimmte rekursive Algorithmen lassen sich in iterative Algorithmen umschreiben und damit effizienter implementieren. Dies trifft insbesondere auf eine sogenannte endständige Rekursion zu, wie z.B. bei der Berechnung der Fakultät.

fak(0)=1; fak(n) = n * fak(n-1);

Rekursive Algorithmen - Fakultät

Das folgende Struktogramm enthält die Umsetzung der rekursiven Definition:



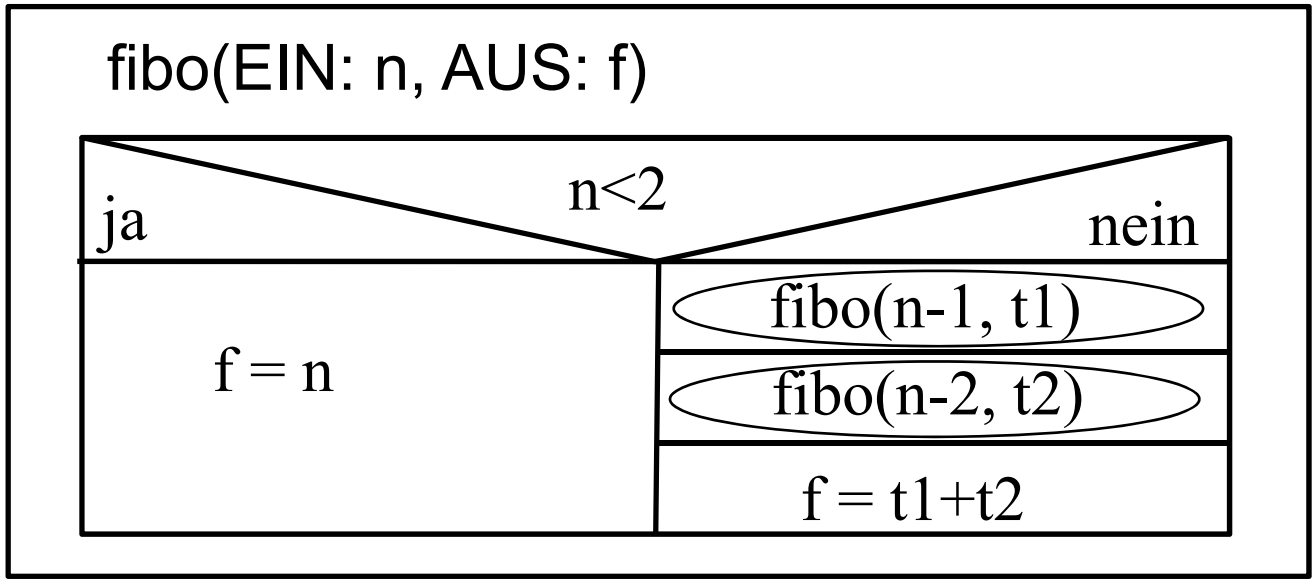
Rekursive Algorithmen – Fibonacci-Zahlen

Rekursive Definition der Fibonacci-Zahlen:

fibo(n) = fibo(n-1) + fibo(n-2)

Anfangsbedingung: fibo(0)=0, fibo(1)=1

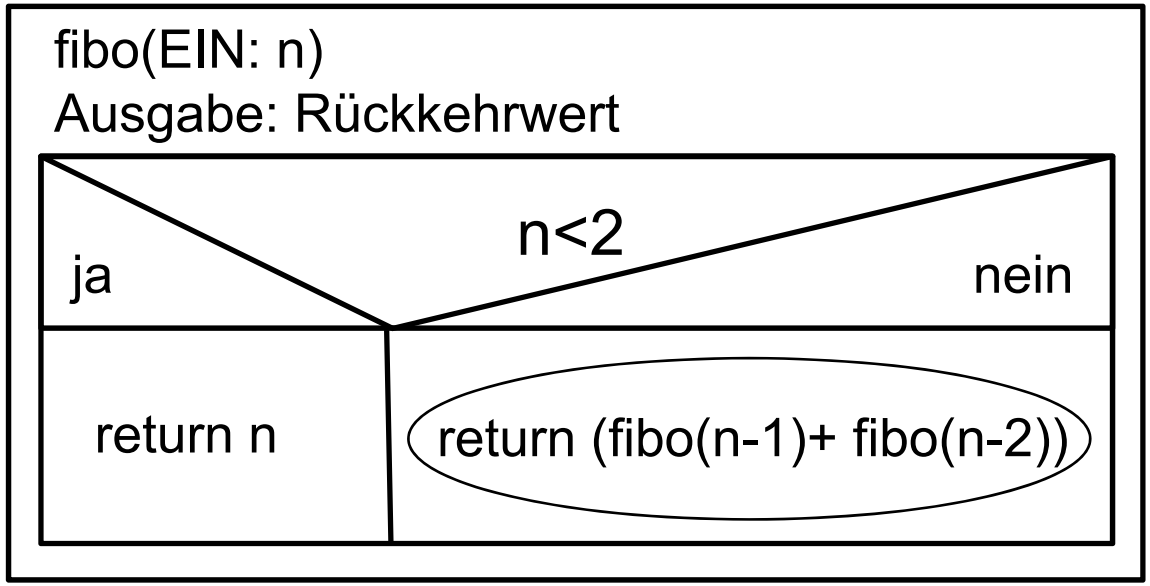
Das folgende Struktogramm enthält die Umsetzung der rekursiven Definition:



Rekursive Algorithmen – Fibonacci-Zahlen

Das folgende Struktogramm benutzt die Rückkehr aus dem Unterprogramm: return

return gibt einen Wert zurück, so dass z.B. x = fibo(5) geschrieben werden kann.



Rekursive Algorithmen – Turm von Hanoi

Das Problem beim Turm von Hanoi besteht in der folgenden Aufgabe:

1. Gegeben ist ein Turm auf einem Standplatz A aus n Scheiben, die übereinander liegen, und zwar immer eine kleinere auf einer größeren Scheibe.
2. Der Turm soll auf einen zweiten Platz B umgesetzt werden, wobei aber beim Umsetzen immer nur eine kleinere auf eine größere Scheibe gelegt werden darf.
3. Bei der Umsetzung darf ein dritter Hilfsplatz C mitbenutzt werden.

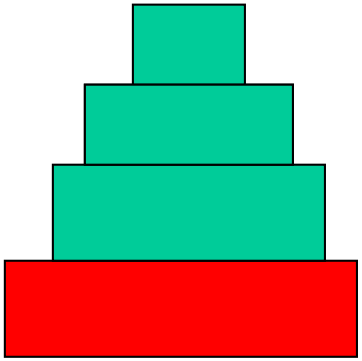
Das C-Programm für dieses Problem wird in der Vorlesung vorgeführt. Es dient als Experimentierprogramm für einen Turm mit einer wählbaren Scheibenanzahl zum Studium der Aufgabenstellung.

Rekursive Algorithmen – Turm von Hanoi

Analysiert man das Problem beim Turm von Hanoi so erkennt man, dass man beim Umsetzen des Turms von n Scheiben vom Platz A zum Platz B erst einmal den Turm von n-1 Scheiben über der größten Scheibe von A nach dem Hilfsplatz C umsetzen muss, um einen Zug der größten Scheibe vom Platz A zum Platz B vornehmen zu können. Danach muss der Turm von n-1 Scheiben vom Platz C wieder auf den Platz B umgesetzt werden.

Rekursive Algorithmen – Turm von Hanoi

Platz A Platz B Platz C



Ausgangssituation

Turm soll nach Platz B umgesetzt werden

Rekursive Algorithmen – Turm von Hanoi

Platz A Platz B Platz C

Turm mit n-1 Scheiben über der größten Scheibe muss auf Hilfsplatz C umgesetzt werden.

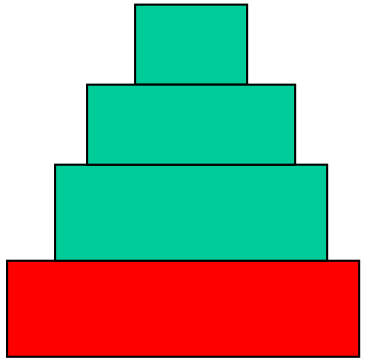
Rekursive Algorithmen – Turm von Hanoi

Platz A Platz B Platz C

Größte Scheibe kann jetzt durch einen Zug vom Platz A zum Platz B befördert werden.

Rekursive Algorithmen – Turm von Hanoi

Platz A Platz B Platz C



Turm mit n-1 Scheiben kann jetzt vom Hilfsplatz C zum Platz B umgesetzt werden.

Rekursive Algorithmen – Turm von Hanoi

Algorithmus rekursiv:

umsetz(n,a,b) =

umsetz(n-1,a,h) , *zug (n,a,b)*, *umsetz(n-1,h,b)*

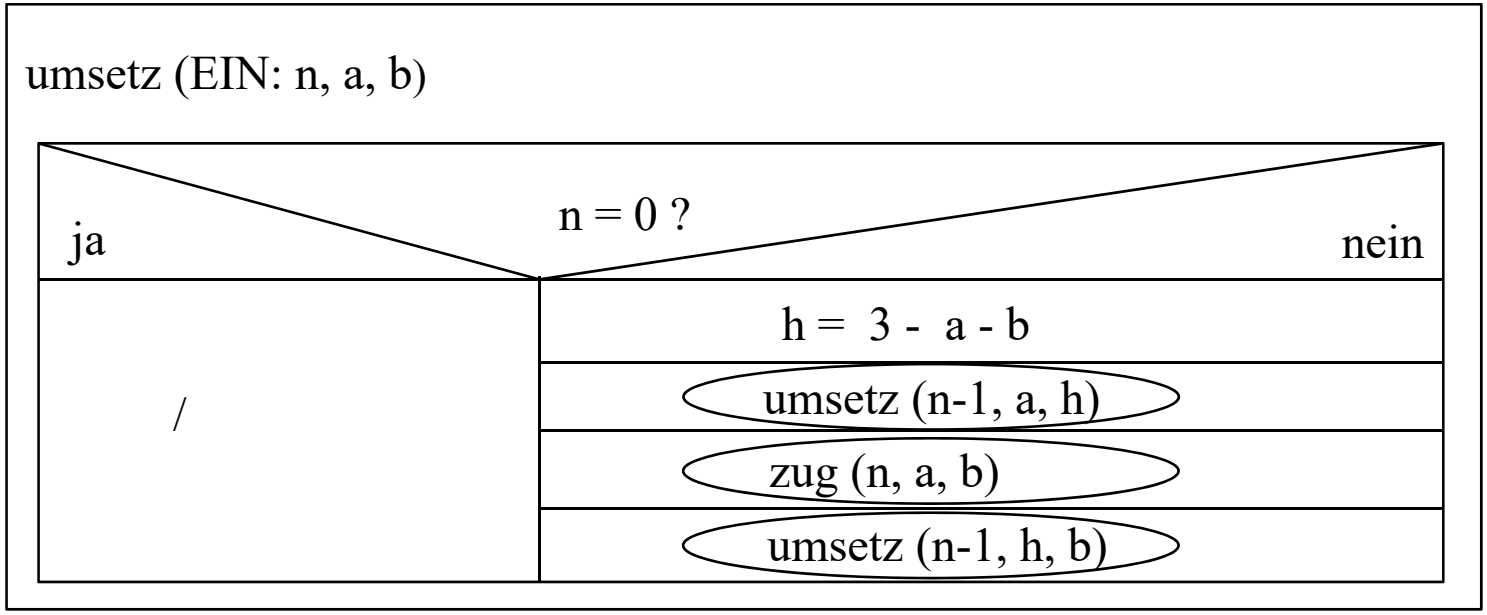
Die Rolle des Hilfsplatzes *h* wechselt von Ebene zu Ebene. Hilfsplatz ist immer der Platz, der in der Umsetzung nicht genannt ist. Bei *umsetz(...,A,C)* ist es in der nächsten Ebene der Platz B. Bei *umsetz(…, C,B)* ist der Hilfsplatz A, usw.

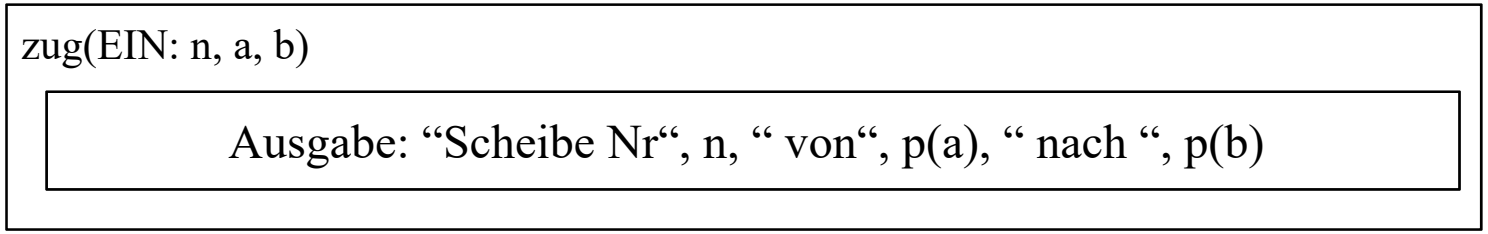
Beziffert man den Platz *A* mit der Ziffer 0, den Platz *B* mit der Ziffer 1, und den Platz *C* mit der Ziffer 2, so kann der freie Platz immer durch *3-a-b* berechnet werden.

Hinweis: a,b,h sind Variable, die jeweils einen Platz (A,B oder C) als Wert aufnehmen.

Rekursive Algorithmen – Turm von Hanoi

Struktogramme rekursiv:



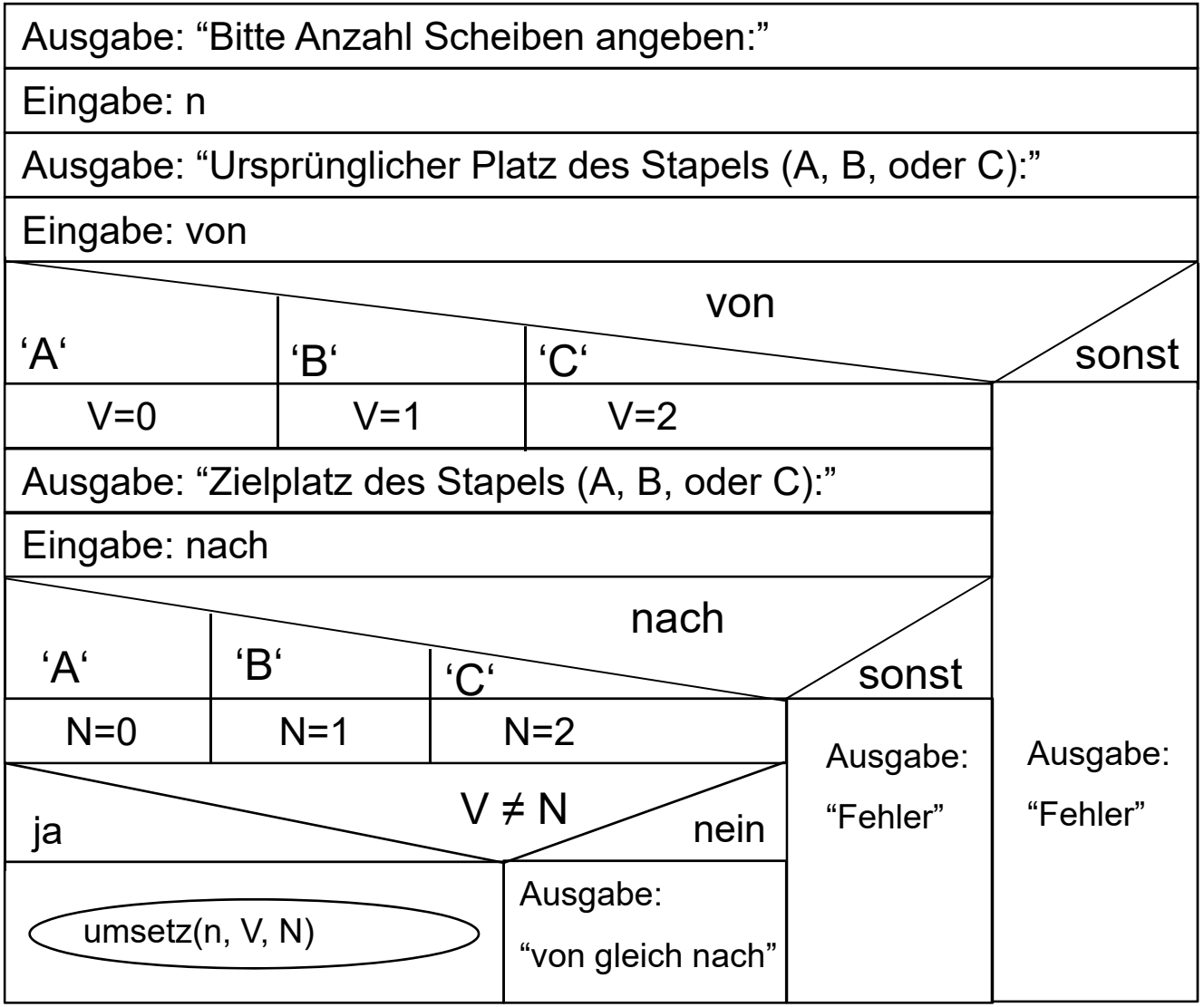


Die Ausgabe benutzt ein vorinitialisiertes Feld p:

‘A‘	‘B‘	‘C‘

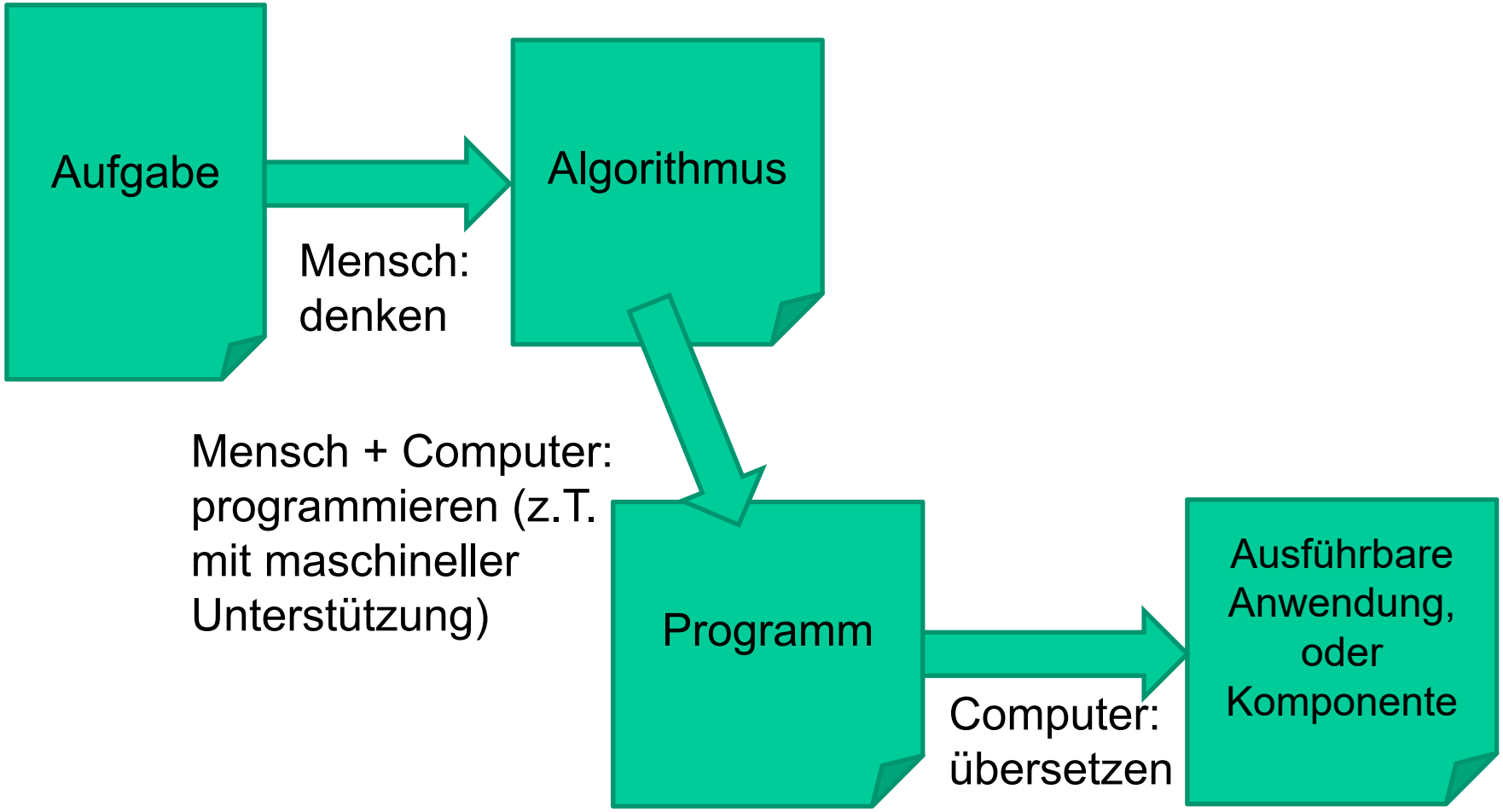
Rekursive Algorithmen – Turm von Hanoi

Hauptprogramm:



Überleitung zu Programmierung (1)

Typisches Vorgehen:



Überleitung zu Programmierung (2)

Notationen für Algorithmen:

- Text, stärker formalisiert
- PAP
- Struktogramm
- Pseudocode
  (allgemeine Programmiersprachenkonstrukte + mathematische Notation)

Überleitung zu Programmierung (3)

Notationen für Programme / Imperative Programmiersprachen:

- FORTRAN: naturwissenschaftliche Berechnungen (sehr schnell)
- COBOL: ökonomische Berechnungen, auch heute noch im Einsatz
- **BASIC: ehemals Anfängersprache,**
  - **heute stark von Microsoft eingesetzt**
  - **VBA-Skript innerhalb Excel, Access**
  - **Visual Basic als VB.Net**
- PASCAL: sehr gut strukturierte Sprache
- C: stark auf Hardware und Betriebssystem bezogen (Mikrocontroller)
- C++, C#, Objective-C: objektorientierte Erweiterungen von C
- JAVA: rechnerunabhängige Sprache mit starkem Internetbezug
- JavaScript: Java-ähnliche Sprache zur Einbettung in Webseiten
- PHP: C-ähnliche Sprache für Webserver-Programmierung