Nittwoch, 8, Februar 2023 05:42

```
Definition
                     -Ein Algorithmus ist ein Verfahren mit einer präzisen endlichen Beschreibung unter Verwendung effektiver Verarbeitungsschritte
Algorithmus
                     • Verfahren: - besteht aus elementaren Verarbeitungsschritten
                                - die einer kontextbedingten Interpretation bedürfen
                                - muss daher am ausführenden Objekt und dessen Möglichkeiten orientiert sein
                     • präzise: - in einer genau festgelegten Sprache abgefasst, die für das ausführende Obiekt verständlich ist
                             - es muss für jeden Verarbeitungsschritt klar sein, welche Tätigkeit auszuführen ist und wann sie durchgeführt werden muss
                      \bullet \text{endlich: $-$> } \text{statisch: die Beschreibung muss endlich sein endlich viele Verarbeitungsschritte} \\
                             -> dynamisch: die Ausführung muss endlich sein
                               - der Algorithmus terminiert
                                - man spricht aber auch von nichtterminierenden "Algorithmen"
                     •Verarbeitungsschritte: - die einzelnen Verarbeitungsschritte müssen tatsächlich ausgeführt werden können
                     =>Ein Algorithmus ist ein Verfahren mit einer präzisen (d.h. in einer eindeutigen Sprache abgefassten) endlichen Beschreibung unter
                       Verwendung effektiver (im Sinne von tatsächlich ausführbarer) Verarbeitungsschritte.
                      >Löst typischerweise nicht genau eine Aufgabe, sondern eine Klasse von Aufgaben
                          -Addiere 3 und 4 vs. Addiere zwei natürliche Zahlen
                          -Welche Aufgabe bearbeitet werden soll, wird durch Parameter bestimmt.
                          -Parameter dienen als Eingabe
                          -Algorithmus liefert für geeignete Eingabe in der Regel ein Ergebnis.
                     •Funktionale Sicht: - Was soll der Algorithmus machen?
•Operationale Sicht: - Wie wird die Aufgabe bewältigt?
                                       - Dazu gehören: -> Elementare Verarbeitungsschritte
                                                       -> Beschreibung der Auswahl der einzelnen auszuführenden Schritte (Kontrollfluss)
                                                       -> Daten und Parameter, die bearbeitet werde
                     SortiereDurchEinfuegen(A)
Algorithmische
                                                                        Branchen: * Verarbeilungsschrift
Sprachelemente
                                → Variablen zur Zuweisung
                                                                               - Konstanten wie n
                                  vertausche die Einträge A[j] und A[j=1]
                                                                          → Konstruckte für Kontrollfluss (Schleifen △)
                               j := j=1ori/hrmikischir Ausdruck
                     •Datenspeicher und Zuweisungen: -Variablen und Zuweisungen zum Speichern von berechneten (Zwischen-)Ergebnissen
                                                    -Konstanten zum Bezeichnen fester Werte
                                                     -Wichtig: sinnvolle Bezeichnerwahl!
                     •Ausdrücke: -Boolesche Ausdrücke
                                 -Arithmetische Ausdrücke
                     •Sequenzierungsanweisungen: -Blöcke zum Gruppieren von Daten und Anweisungen
                                                  -Funktionsaufrufe zum mehrmaligen Wiederverwenden einmal definierter Algorithmen
-Bedingte Anweisungen zum Verzweigen im Programmfluss
                                                  -Iterationsanweisungen zur Realisierung von Schleif
                     •Hilfskonstrukte: -Kommentare
Sprünge VS.
                      Verzweigung mit Sprüngen:
Strukturierte
Verzweigungen
                             IF Bedingung THEN GOTO marke
                             /* Anweisungen, die ausgeführt werden,
                                 falls Bedingung nicht erfüllt
                      marke: /* Anweisungen, falls Bedingung erfüllt
                      marke2: /* Ende der Verzweigung
                     Strukturierte Verzweigung:
                            IF Bedingung
                                   /* Anweisungen, wenn Bedingung wahr
                                   /* Anweisungen, wenn Bedingung falsch
                            ENDIF
                     Schleife mit Sprüngen:
                     loop: IF NOT (Bedingung) THEN GOTO ende
                                   /* Anweisungen, solange Bedingung wahr
                                         GOTO loop
                      ende:/* Weiter im Programmfluss...
                     Strukturierte Schleifen
                             WHILE Bedingung DO
                             /* Anweisungen, solange Bedingung wahr
                             END WHILE
                             /* Weiter im Programmfluss...
Modularisierung
                     = Zerlegung in Teilprobleme
                     · Einzelteile idealerweise wiederverwendbar
                          -> an mehreren Stellen innerhalb eines Programms
                          -> in anderen Programmen
```

Klassifizierung von Algorithmen

Eigenschaften:

- •terminierend: Algorithmus heißt für eine Eingabe terminierend, wenn er stets nach endlich vielen Schritten endet.
- •deterministisch: Algorithmus heißt für eine Eingabe deterministisch, wenn in der Auswahl der Verarbeitungsschritte keine Freiheit besteht.
- •determiniert: Algorithmus heißt für eine Eingabe determiniert, wenn das Resultat des Algorithmus eindeutig bestimmt ist.
- •sequentiell: Algorithmus heißt für eine Eingabe sequentiell, wenn die Verarbeitungsschritte stets hintereinander ausgeführt werden.
- parallel: Algorithmus heißt für eine Eingabe *parallel*, wenn gewisse Verarbeitungsschritte nebeneinander (gleichzeitig) ausgeführt werden.

->Anmerkung: Viele Definitionen von Algorithmus fordern, dass Algorithmen immer deterministisch sind und/oder dass Algorithmen immer terminieren.

Sortieralgorithmen

SELECTION SORT

Eingabe: unsortierte Liste X

Vorgehen:

- · Erzeuge leere Liste Y
- · wähle kleinstes Element aus der Liste X aus
- · Lösche es aus X
- · Füge es ans Ende von Y an
- · Wenn X leer ist: Gebe Y aus

INSERTION SORT

Eingabe: unsortierte Liste X

Vorgehen:

- 1. Setze Markierung auf zweites Element der Liste
- Vergleiche markiertes Element mit Vorgängerelement:

 - Wenn kleiner: tausche die beiden Wiederhole 2, bis kein Tausch mehr stattfindet
- 3. Verschiebe Markierung um eins nach rechts
- Markierung nach Listenende: fertig
 Sonst: gehe zu 2

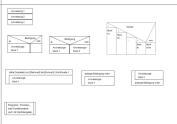
MERGESORT

Eingabe: unsortierte Liste X

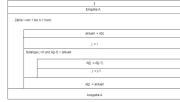
Vorgehen

Rekursiv: zerlege Liste in Teillisten, sortiere Teillisten und mische sie dann rekursiv wieder

NASSI-SHNEIDERMAN-DIAGRAMME ->Struktogramm



-Bsp. Insertionsort



ZYKLENSUCHE IN GERICHTETEM GRAPHEN

- prozedur Zyklensuche (Graph G. Knoten x) falls Markierung(x) = "in Bearbeitung" dann Zyklus gefunden sonst falls Markierung(x) = "noch nicht begonnen" dann
- Markierung(x) := "in Bearbeitung";
 für alle Nachfolgerknoten y von x tue Zyklensuche(y); Markierung(x) := "abgeschlossen"

