

Algorithmen

Riedmann

Inhalt

- Definition
- Eigenschaften
- Darstellung
- Arten von Algorithmen
- Suchalgorithmen

Definition

- „Ein Algorithmus ist im Wesentlichen eine Anleitung zur Lösung eines Problemes“
- Algorithmen treten nicht nur in der Informatik auf sondern begegnen uns täglich
- Beispiel
 - Zusammenbau eines IKEA Kastens
 - Kochrezept
 - Schulweg

Definition

- Oft sind solche Anleitungen nicht präzise genug formuliert um eindeutig gelöst werden zu können
 - Ein korrekter Algorithmus darf keinen Interpretationsspielraum mehr zulassen!
 - Eine „mechanische“ Lösung muss möglich sein!

Aufgabe

- Formuliere einen Algorithmus zur schriftlichen Addition von 2 Zahlen
 - Die Formulierung eines Algorithmus in „Prosa“ kann sehr aufwendig sein
 - Eine genaue Formulierung ist schwierig
 - Verwende Diagramme
 - Mathematik

Eigenschaften eines Algorithmus

- Korrektheit

- Der Algorithmus erfüllt die Spezifikation

- Vollständigkeit

- Die Lösungsbeschreibung muss vollständig sein – es darf nichts fehlen

- Eindeutigkeit

- Es darf keinen Interpretationsspielraum geben

- Effizienz

- Bestmögliche Ressourcenausnutzung

Bestandteile eines Algorithmus

- Datenobjekte und Datentypen
 - Ein Datentyp definiert eine Menge von möglichen Werten
 - Elementare Datentypen
 - Können nicht mehr weiter zerlegt werden (z.B. eine Zahl)
 - Strukturierte Datentypen
 - Haben eine Struktur → Sind aus kleineren Bestandteilen zusammengebaut
 - Array, Listen, Sets

Bestandteile eine Algorithmus

- Anweisungen
 - Zuweisungen
 - Ausdrücke
 - Verzweigungen
 - Wiederholungen

Bestandteile eine Algorithmus

- Anweisungen
 - Zuweisungen
 - Ausdrücke
 - Verzweigungen
 - Wiederholungen

Darstellung von Algorithmen

- Ablaufdiagramm
- Struktogramm
- UML
- Prosa
- Pseudocode
- Programmiersprache

Wichtige Arten von Algorithmen

● Suchalgorithmen

- Wie können Datensätze effizient in Datenbeständen gefunden werden
- Wie müssen die Daten strukturiert sein?

● Sortieralgorithmen

- Daten liegen oft unsortiert in Datenstrukturen vor
- Daten werden aber sortiert benötigt

● Erzeugen von Zufallszahlen

● Mustersuche

● Exhaustionsalgorithmen

- Finden von Lösungen durch probieren

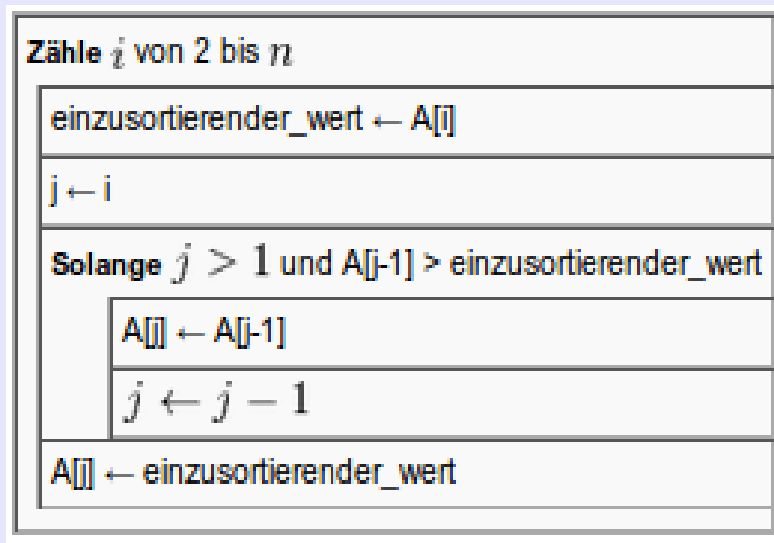
● Sudoku, Schachaufgaben etc.

Sortieralgorithmen

- Insertionsort
- Bubblesort
- Selectionsort

Insertionsort

- Insertion Sort
 - Ähnlich der Sortierung eines Kartenstapels



Quelle: <http://de.wikipedia.org/wiki/Insertionsort>

Insertionsort

λ Laufzeitverhalten

Quelle: <http://de.wikipedia.org/wiki/Insertionsort>

Selectionsort

- Es wird das kleinste Element gesucht und an die erste Stelle gesetzt
- Danach wird das restliche Array durchsucht und wieder die kleinste Zahl nach vorne gesetzt

4 2 1 6 3 5	Das Minimum ist 1. Vertausche also das 1. und das 3. Element.
1 2 4 6 3 5	Das Minimum des rechten Teilarrays ist 2. Da es bereits an 2. Position steht, wird es nicht getauscht.
1 2 4 6 3 5	Wir haben jetzt bereits ein sortiertes Teilarray der Länge 2. Wir vertauschen nun 4 und das Minimum 3.
1 2 3 6 4 5	Wir vertauschen 6 und 4.
1 2 3 4 6 5	Wir vertauschen 6 und 5.
1 2 3 4 5 6	Das Array ist jetzt fertig sortiert.

Selectionsort

- Laufzeitverhalten

Bubblesort

- Paarweises vergleichen von links nach rechts
- Nach dem ersten Durchlauf ist das kleinste Element ganz links

```
55 07 78 12 42  1. Durchlauf
07 55 78 12 42
07 55 78 12 42
07 55 12 78 42  Letzter Vergleich
07 55 12 42 78  2. Durchlauf
07 55 12 42 78
07 12 55 42 78  Letzter Vergleich
07 12 42 55 78  3. Durchlauf
07 12 42 55 78  Letzter Vergleich
07 12 42 55 78  4. Durchlauf + Letzter Vergleich
07 12 42 55 78  Fertig sortiert.
```

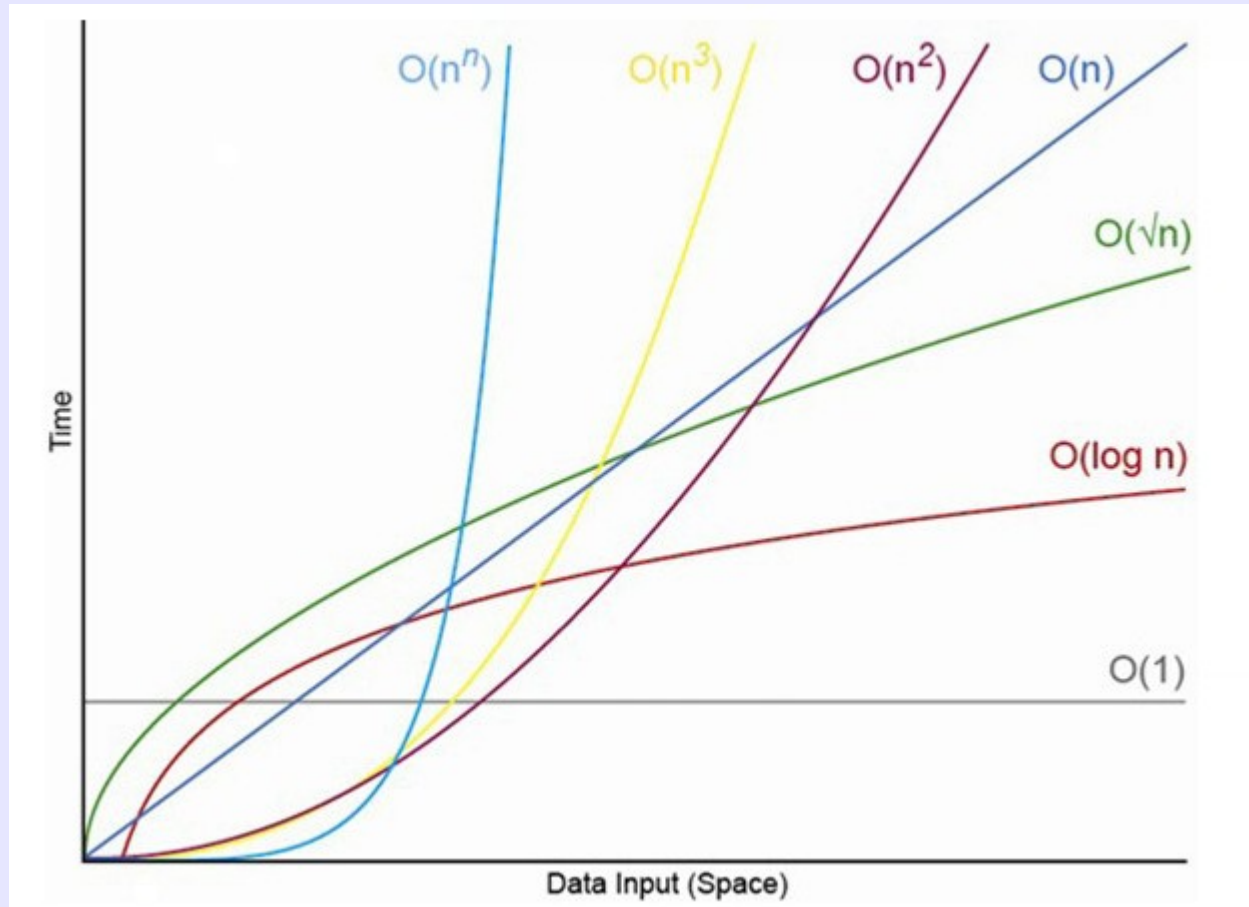
Bubblesort

- Laufzeitverhalten:

Die O-Notation

- Die Effizienz eines Algorithmus wird mit der “Big O Notation” dargestellt
- Dies ist eine mathematische Darstellung des Laufzeitverhaltens

Die O-Notation



Die O-Notation

$\lambda O(1)$

□ Performance ist Konstant

$\lambda O(n)$

□ Performance ist linear

$\lambda O(n^2)$

□ Performance ist direkt proportional mit der Datenmenge

$\lambda O(n \log n)$

□ Performance ist logarithmisch. Bei kleinen Datenmengen oft schlechter, bei großen besser als $O(n)$