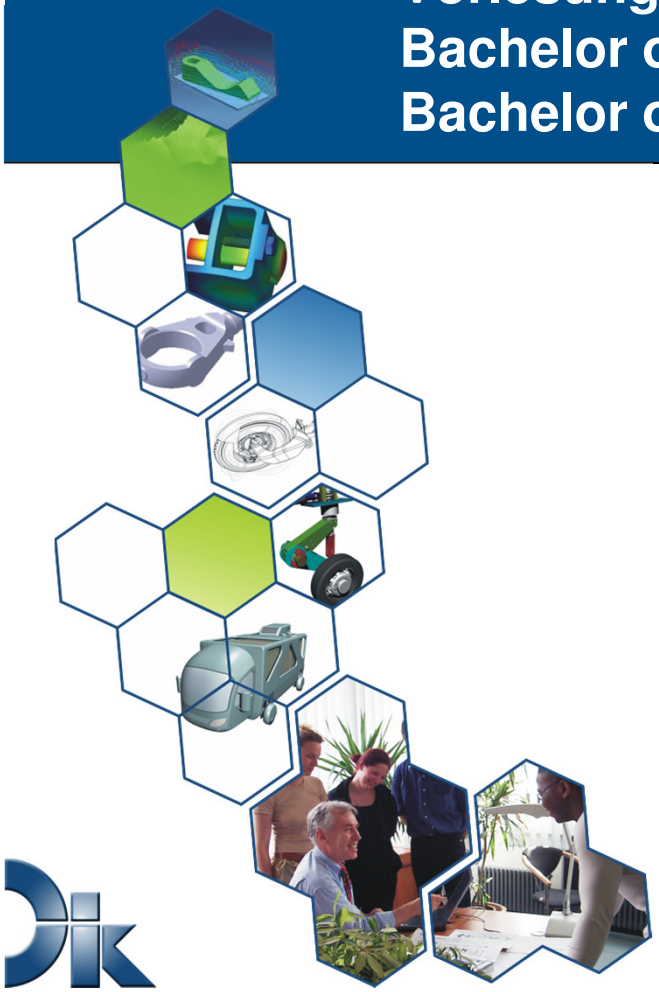
Grundlagen der elektronischen Datenverarbeitung

Vorlesung für Bachelor of Science MPE, Mechanik, Bachelor of Education

TECHNISCHE UNIVERSITÄT DARMSTADT



Die Vorlesung wird zusätzlich in den Raum S3|11-006 (1 Stockwerk tiefer) übertragen.

Gliederung 8. Vorlesungsstunde GeDV

TECHNISCHE UNIVERSITÄT DARMSTADT

- Abstrakte Datentypen
  - Binärer Baum
- Algorithmen
  - Komplexität & Ordnungsklassen
  - Algorithmenklassen
- Mathematische und technische Grundlagen
  - Zahlendarstellungen
  - Komplement

Abstrakte Datentyp: Baum (Wdh.)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Ein Baum ist ein gerichteter Graph mit Knoten, auf die, mit Ausnahme der Wurzel, jeweils nur eine Kante zeigt.
- Die Wurzel besitzt keine eingehende Kante.
- Von den Knoten des Baumes weisen keine, eine oder mehrere Kanten zu weiteren Knoten des Baumes. Knoten ohne ausgehende Kanten heißen Blätter, alle anderen Knoten heißen innere Knoten.
- Ein Knoten und alle über seine Kanten referenzierbaren Knoten heißen zusammengenommen *Teilbaum*.
- Die Anzahl der Kanten von der Wurzel zu einem Knoten heißt Weglänge des Knotens.
- Die maximale Weglänge über alle Knoten heißt „Tiefe des Baumes“.
- Wesentliche Baumtypen sind
  - Bäume mit einer Ausgangskante je Knote (einfach verketteten Listen),
  - Bäume mit zwei Ausgangskanten je Knoten (binäre Bäume) und
  - Bäume mit mehr als zwei ausgehenden Kanten je Knoten (Vielweg- oder B Bäume).

Skript S. 102 ff

Arten von Bäumen (Wdh.)

Skript S. 102 ff

Abstrakter Datentyp: Baum (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Jeder Knoten eines Baumes besitzt einen Schlüssel, nach dem der Knoten in den Baum eingefügt wird.
- Auf der Schlüsselmenge existiert eine Ordnungsrelation, z.B. *kleinergleich*, die als Entscheidungskriterium für den Ort des Einfügens dient.
- Gebräuchliche Operationen auf Bäume sind:
  - Erzeugen eines Baumknotens,
  - Einfügen eines Baumknotens anhand des Schlüssels und der Ordnungsrelation,
  - Löschen eines Baumknotens,
  - Suchen nach einem Knoten anhand eines vorgegebenen Schlüssels und
  - Traversieren eines Baumes (Besuchen jedes Knotens verbunden mit dem Ausführen einer Operation auf den Nutzdaten des Knotens).

Skript S. 102 ff

Abstrakte Datentypen: Binärer Baum (1)

TECHNISCHE UNIVERSITÄT DARMSTADT

Für das **Einfügen eines Knotens** in einen binären Baum gibt es verschiedene Möglichkeiten:

- **Naturwüchsiger Baum**: Neue Knoten werden stets als Blätter an den Baum gehängt. Bei auf- oder absteigend vorsortierten Folgen von einzufügenden Knoten kann dies jedoch zu einer Degenerierung des Baumes in eine einfach verkettete Liste führen.

- **Vollständig ausgeglichener Baum**: Knoten können hier auch als innere Knoten eingefügt werden. Da aber jeder vollständig ausgeglichene Baum zwei Teilbäume gleicher Tiefe besitzt, ist der Aufwand nach dem Einfügen eines Knotens zur Wiederherstellung dieser Bedingung sehr hoch. Der Suchaufwand ist jedoch außerordentlich gering.

- **Ausgeglichener Baum (AVL-Baum)**: Die Tiefen der Teilbäume eines Knotens dürfen sich um 1 unterscheiden, was den Aufwand zur Wiederherstellung der Ausgeglichenheitsbedingung erheblich reduziert. Der Suchaufwand erhöht sich jedoch nur um einen Knoten. Der AVL-Baum ist der am weitesten verbreitete Baum.

Skript S. 103 ff

Abstrakte Datentypen: Binärer Baum (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Beim **Suchen in einem binären Baum** nach einem Knoten mit einem bestimmten Schlüssel sind von der Wurzel ausgehend folgenden Fragen zu beantworten:

(1) Stimmen der vorgegebene Schlüssel und der Schlüssel des aktuell besuchten Knotens überein? Wenn ja, so ist der Knoten gefunden.

(2) Ist der vorgegebene Schlüssel entsprechend der Ordnungsrelation kleiner als der Knotenschlüssel? Wenn ja, befindet sich der gesuchte Knoten, wenn überhaupt, im linken Teilbaum des aktuellen Knotens. Wende dann die Fragen auf den linken Nachfolger des aktuellen Knotens an, falls dieser existiert. Falls er nicht existiert, ist der gesuchte Knoten nicht im Baum.

(3) Andernfalls befindet sich der gesuchte Knoten, wenn er überhaupt in den Baum eingefügt wurde, im rechten Teilbaum des aktuellen Knotens. Wende daher die Fragen auf den rechten Nachfolger des aktuellen Knotens an, falls dieser existiert. Falls er nicht existiert, ist der gesuchte Knoten nicht im Baum

Skript S. 103 ff

Besondere Graphen: Baum, binärer Baum

Demo: Einfügen in AVL-Baum

Abstrakte Datentypen: Binärer Baum (3)

TECHNISCHE UNIVERSITÄT DARMSTADT

Das **Traversieren eines binären Baumes** geht von der Wurzel des Baumes aus und kann auf drei Arten erfolgen:

- **Präorder-Traversierung**:
  Zunächst wird die *Operation* auf dem aktuellen Knoten ausgeführt, dann der *linke* Teilbaum und danach der *rechte* Teilbaum traversiert.

- **Inorder-Traversierung**:
  Zunächst wird der *linke* Teilbaum des aktuellen Knotens traversiert, dann die *Operation* auf dem Knoten ausgeführt und danach der *rechte* Teilbaum traversiert.

- **Postorder-Traversierung**:
  Zunächst wird der *linke* Teilbaum des aktuellen Knotens traversiert, dann der *rechte* Teilbaum und danach wird die *Operation* auf dem Knoten ausgeführt.

Skript S. 104

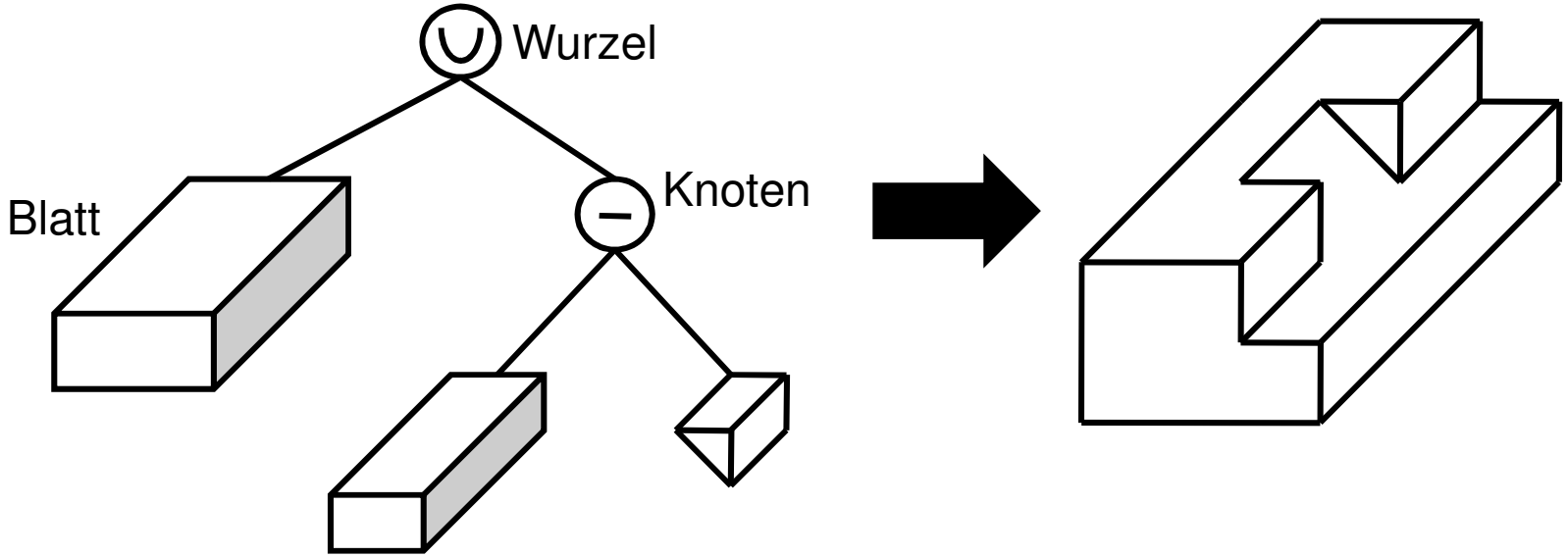
Beispiel: Traversieren binärer Bäume

CSG-Binärbaum

TECHNISCHE UNIVERSITÄT DARMSTADT

CSG (Constructive Solid Geometry)

- Konstruieren von komplexen Körpern aus Formprimitiven durch Mengenoperationen
- Die Blätter im binären Baum enthalten die Formprimitive, die Knoten enthalten die Mengensymbole.



Kapitel 4.2

Algorithmen

Algorithmen

TECHNISCHE UNIVERSITÄT DARMSTADT

Entwicklung von Algorithmen in der Entwurfsphase

Algorithmen

TECHNISCHE UNIVERSITÄT DARMSTADT

Spezifikation

- Zur Beschreibung einer Methode, der Spezifikation, wird ihre *Signatur* und ihre *Semantik* verwendet.

Signatur

- Die Signatur einer Methode besteht aus der Zusammenstellung der Datentypen, ihrer Ein- und Ausgabeparameter und deren Reihenfolge.

Semantik

- Die Semantik einer Methode ist eine Beschreibung, was die Methode leistet (Nachbedingung), wenn die Eingabeparameter bestimmte Eigenschaften erfüllen (Vorbedingung).

Algorithmus

- Ein Algorithmus ist ein Verfahren zur Lösung einer Aufgabe, wie sie beispielsweise durch die Signatur und Semantik einer Methode gestellt wird. Ein Algorithmus beschreibt also, wie die Methode das leistet, was sie nach ihrer Semantik leisten soll.

Skript S. 109 ff

Entwicklung von Algorithmen in der Implementierungsphase

TECHNISCHE UNIVERSITÄT DARMSTADT

Komplexität

TECHNISCHE UNIVERSITÄT DARMSTADT

Komplexität

- Die Komplexität eines Algorithmus ist der Aufwand, den Algorithmus auf einem Rechner oder per Hand auszuführen.

Problem- oder Aufgabengröße

- Die Komplexität hängt vom Umfang der zu bearbeitenden Daten, ab. Dieser Umfang wird Problemgröße genannt.

Kostenmaß

- Zur Quantifizierung der Komplexität wird ein abstraktes Kostenmaß definiert.

Zeitkomplexität

- Bezieht sich das Kostenmaß auf die Ausführungszeit des Algorithmus, so wird die Komplexität Zeitkomplexität genannt.

Platzkomplexität

- Bezieht sich das Kostenmaß auf den Speicherplatzbedarf des Algorithmus, so wird die Komplexität Platzkomplexität genannt.

Skript S. 110 ff

Ordnungsklassen

TECHNISCHE UNIVERSITÄT DARMSTADT

O-Notation

Komplexitäten von Algorithmen können in Klassen eingeteilt werden, deren Mitglieder für große Probleme einen vergleichbaren Aufwand verursachen.

Zur Bildung der Klassen wird die O-Notation verwendet

$$f(n) = O(g(n))$$ („*f ist von der Ordnung g*“)

Ordnungsklassen, denen eine Komplexität eineindeutig zugeordnet werden kann, sind (in aufsteigender Komplexität):

$$\log(\log n), \log n, n, n^2, n^3$$

Skript S. 113 f

Zeitkomplexitäten (1)

TECHNISCHE UNIVERSITÄT DARMSTADT

konstante Komplexität: O(n)=1

Logarith. Komplexität: O(n)=log n

Beispiel:

- Suchen nach einem Element in einer Liste der Länge n, etwa in einem Telefonbuch

Lineare Komplexität: O(n)=n

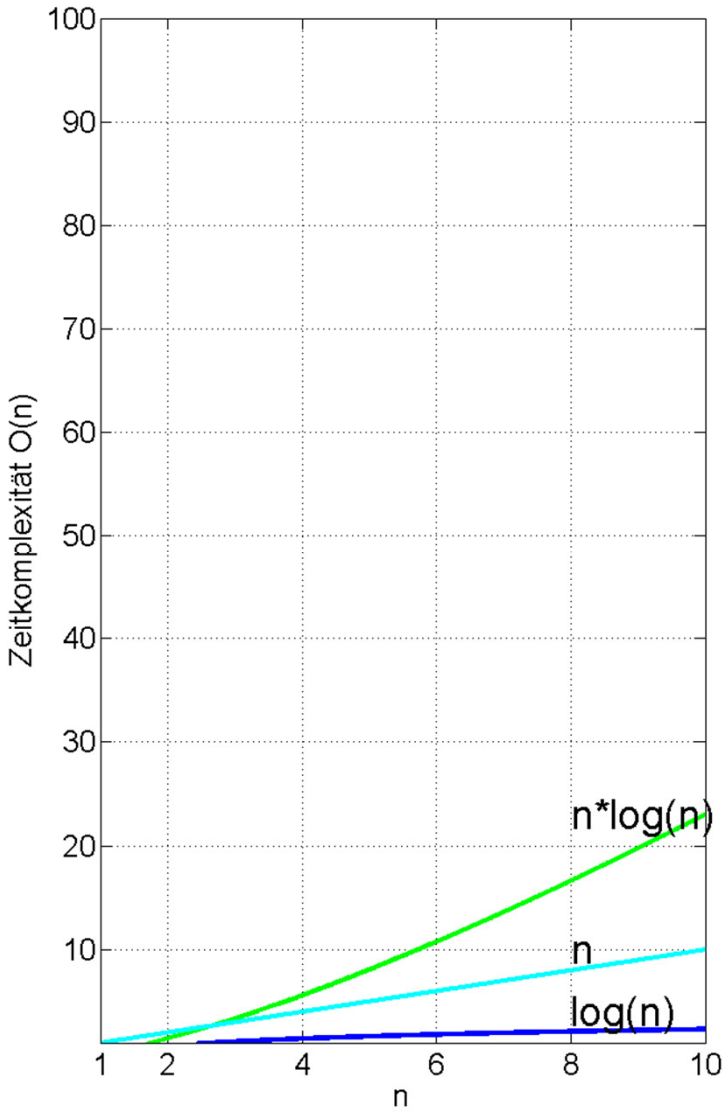
Beispiele:

- Suchen in einer Liste der Länge n
- Addition, Subtraktion und Skalarprodukt-bildung bei n-dim. Vektoren
- Verschlüsseln einer Nachricht der Länge n

n*log(n)-Komplexität : O(n)=n*log(n)

Beispiele:

- Sortieren von Listen der Länge n mit dem Quicksort
- Suchen in einem binären Baum der Tiefe n



Zeitkomplexitäten (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

Quadrat. Komplexität: $O(n)=n^2$

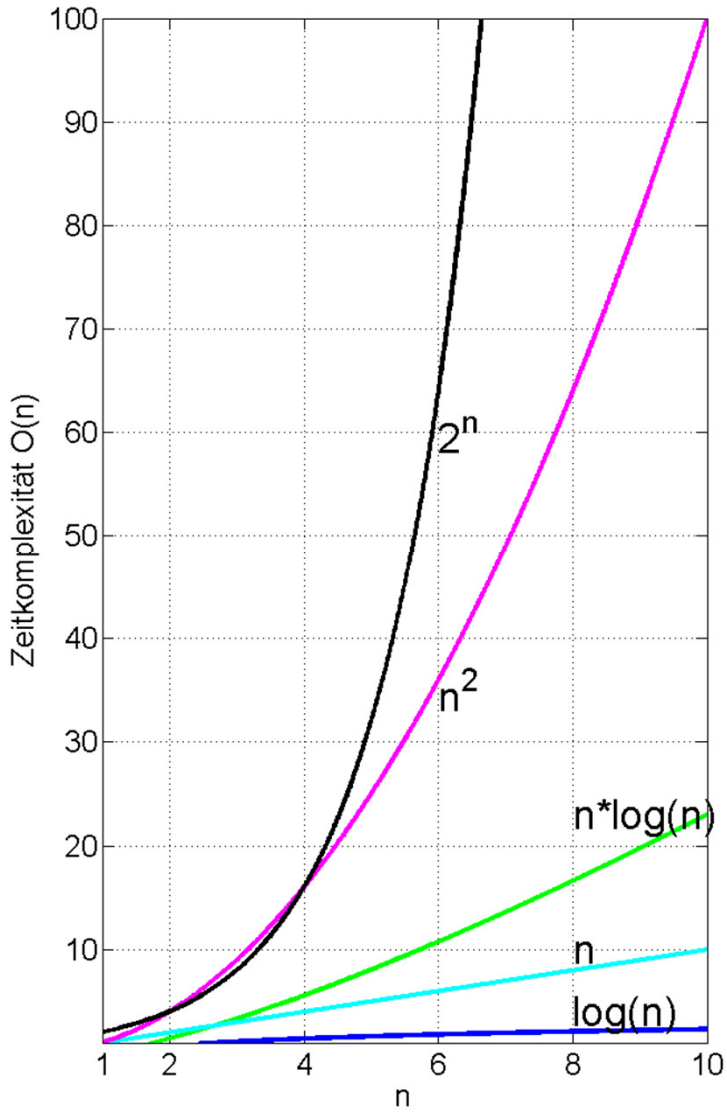
Beispiele:

- Sortieren von Listen der Länge n mit dem Bubblesort
- Multiplikation einer Matrix mit einem Vektor der Länge n

Exponent. Komplexität: $O(n)=2^n$

Beispiel:

- Entschlüsseln einer verschlüsselten Nachricht der Länge n



Beispiel zur O-Notation: Diskretisierung eines Würfels

Algorithmenklassen

TECHNISCHE UNIVERSITÄT DARMSTADT

Algorithmen können nach unterschiedlichen Kriterien in Klassen aufgeteilt werden:

- **Alle Algorithmen, die ein bestimmtes Problem lösen**
  - Sortieralgorithmen
  - Suchalgorithmen

- **Alle Algorithmen, die eine bestimmte Methode zur Problemlösung verwenden:**
  -Rekursive Algorithmen

- **Alle Algorithmen, die eine bestimmte Klasse von Problemen lösen**
  - Graphenalgorithmen
  - Kryptographische Algorithmen
  - Kodierungsalgorithmen
  - Numerische Algorithmen

Skript S. 114 ff

Algorithmen

TECHNISCHE UNIVERSITÄT DARMSTADT

Die wichtigsten Vertreter der Algorithmen sind:

- **Sortieralgorithmen**
- **Suchalgorithmen**
- **Rekursive Algorithmen**
- **Graphenalgorithmen**
- **Kryptographische- und Kodierungsalgorithmen**
- **Numerische Algorithmen**

Skript S. 114 ff

Sortieralgorithmen

TECHNISCHE UNIVERSITÄT DARMSTADT

Sortieralgorithmen

- können bestimmte Objekte (z.B. Daten, Zahlen…) die in einer Ordnungsrelation definiert sind, in eine bestimme Reihenfolge bringen.
- Die Ordnungsrelation setzt jeweils zwei Objekte einer Menge M der zu sortierenden Objekte in eine Beziehung.
- Auswahl des Sortieralgorithmus ist Abhängig von der Objektmenge.

Sortieren von Feldern entsprechend der Ordnungsrelation

- Einfügen eines neuen Elements
- Auswählen eines Elements
- Austauchen von Elementen

→ Beispiele:
Quicksort: O(n)=n*log(n),
Aufteilen des Problems in Teilprobleme (divide and conquer);
Bubblesort: $O(n)=n^2$
…

Beispiel: Bubblesort Algorithmus in MATLAB

TECHNISCHE UNIVERSITÄT DARMSTADT

```
function Y= bubbleSort(X)
n=length(X);
for ii = 1:n-1
    for jj = n-1:-1:ii
        if X(jj)>X(jj+1)
            zw = X(jj);
            X(jj) = X(jj+1);
            X(jj+1) = zw;
        end
    end
end
Y=X;
```

Beispiel: X = „BEISPIEL“

1. BEIPIELS
2. BEIIELPS
3. BEIEILPS
4. BEEIILPS

Beispiel: Ergebnisentstehung mit Sortieren nach Bubblesort

Demo: Sortieralgorithmen

Suchalgorithmen

TECHNISCHE UNIVERSITÄT DARMSTADT

Suchbedingung

- Ähnlich einer Ordnungsrelation für das Sortieren stellt eine Suchbedingung fest, ob ein zu testendes Objekt mit dem gesuchten Objekt übereinstimmt.

Suchen auf sequentiellen Dateien, Feldern und Bäumen

- Sequentielles Suchen
- Binäres Suchen
- Tabellen-Suchen

Suchen in Zeichenketten (Texten)

- Direkte Mustersuche
- Knuth-Morris-Pratt-Algorithmus
- Boyer-Moore-Algorithmus

Rekursive Algorithmen

TECHNISCHE UNIVERSITÄT DARMSTADT

Rekursion

- Ein Algorithmus ist rekursiv, wenn er das gestellte Problem in ein oder mehrere Teilprobleme aufteilt und sich selbst zur Lösung dieser Teilprobleme verwendet.
- Die Algorithmen A und B sind indirekt rekursiv, wenn der Algorithmus A den Algorithmus B zur Lösung eines Teilproblems verwendet und B wiederum A benutzt.

Abbruchbedingung

- Ein rekursiver Algorithmus darf sich nicht immer selbst aufrufen, da sonst die Rekursion beliebig fortschreitet. Daher besitzt der Algorithmus eine Abbruchbedingung, die in manchen Fällen erfüllt ist und in die Rekursion abbricht.

Monotones Verhalten

- Das Verhalten eines rekursiven Algorithmus´ muß so gestaltet sein, dass die Abbruchbedingung in endlicher Zeit erfüllt werden kann.

Skript S. 117

Kryptographische Algorithmen

TECHNISCHE UNIVERSITÄT DARMSTADT

Ziele der Kryptographie (u.a.)

- Nachrichten vor Lauschangriffen durch Verschlüsselung (Chiffrierung) schützen und
- dem Empfänger der Nachricht ermöglichen, festzustellen, ob die Nachricht in der Tat von dem erwarteten Sender kommt oder durch einen Fälscher eingespielt wurde (Authentifizierung)

Algorithmen zur Chiffrierung und Dechiffrierung

- Algorithmen für sequentielle Chiffren mit nichtlinearen Pseudozufallszahl-Generatoren
- Algorithmen für Blockchiffren
- Algorithmen für Chiffren aus Einwegfunktionen

Algorithmen zur Authentisierung

- Private Schlüsselsysteme
- Öffentliche Schlüsselsysteme

Skript S. 119 ff

Graphenalgorithmen

TECHNISCHE UNIVERSITÄT DARMSTADT

Allgemeine Algorithmen auf Graphen

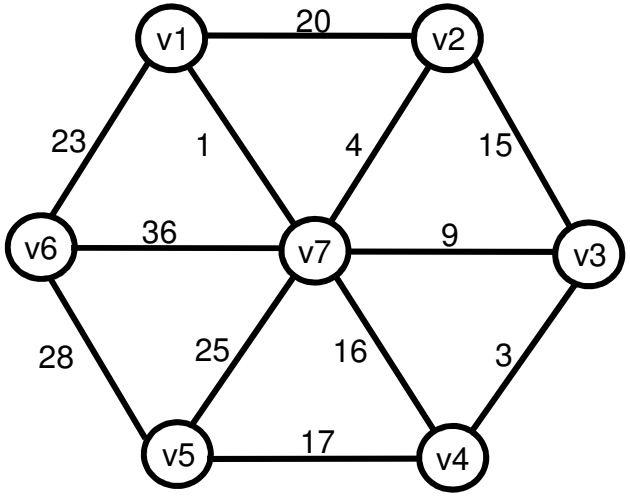
- Tiefensuche
- Breitensuche

Algorithmen auf gewichteten Graphen

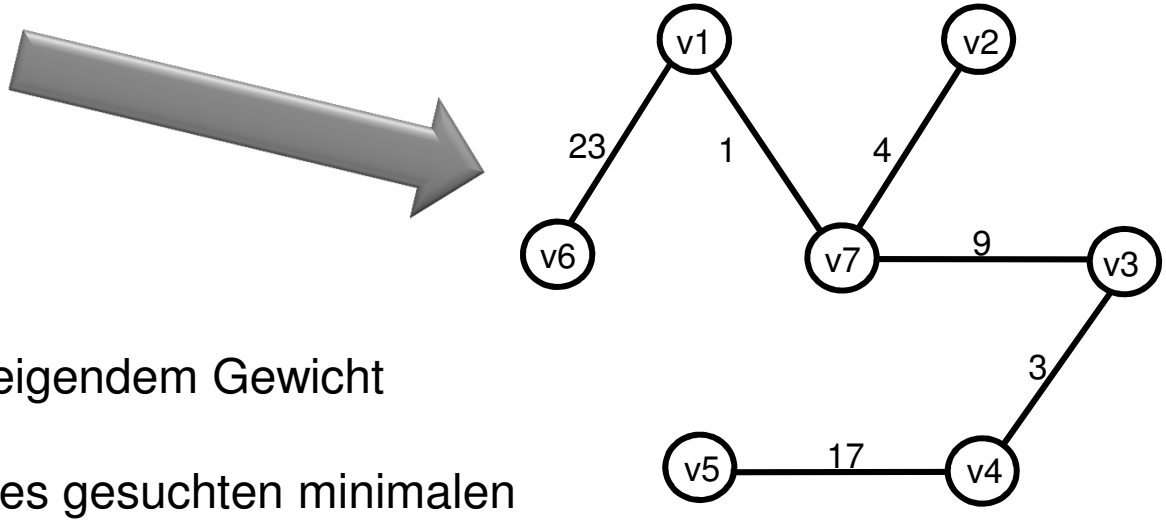
- Suchen des kürzesten Pfades zwischen zwei Knoten (die Summe der Kantengewichte ist minimal)
- Aufstellen des minimalen Spannbaumes (Reduktion der Kantenmenge auf die Kanten, die alle Knoten miteinander verbinden, daß die Summe der Kantengewichte zwischen zwei beliebigen Knoten minimal ist)

Skript S. 118 f

Beispiel: Minimaler Spannbaum



Ein minimaler Spannbaum eines Graphen ist eine Kantenmenge, die alle Knoten so verbindet, daß ein Baum entsteht und die *Summe der Kanten minimal* ist.



Algorithmus von Kruskal

1. Sortiere Kanten nach aufsteigendem Gewicht
   Ergebnis: Kantenliste L
2. Initialisiere Kantenmenge des gesuchten minimalen Spannbaums durch die leere Menge: S={}
3. Füge aus der Kantenliste L eine Kante nach der anderen der Menge S nur dann hinzu, falls das Ergebnis S nach wie vor einen Baum darstellt.

Skript S. 118 f

Beispiel: Minimaler Spannbaum

Numerische Algorithmen

TECHNISCHE UNIVERSITÄT DARMSTADT

Numerische Algorithmen sind Berechnungsverfahren zur Lösung mathematischer Probleme.

Klassen von Numerischen Algorithmen:

- Numerische Verfahren zur Lösung algebraischer und transzendentaler Gleichungen
- Verfahren zur numerischen Lösung linearer Gleichungssysteme
- Verfahren zur numerischen Lösung nichtlinearer Gleichungssysteme
- Eigenwerte und Eigenvektoren von Matrizen
- Approximation stetiger Funktionen
- Interpolation und Splines
- Numerische Differentiation
- Numerische Integration
- Numerische Verfahren für Anfangswertprobleme bei gewöhnlichen Differentialgleichungen erster Ordnung
- Numerische Verfahren für Anfangswertprobleme bei Systemen von gewöhnlichen Differentialgleichungen erster Ordnung und bei Differentialgleichungen höherer Ordnung

Skript S. 122 ff

Kapitel 5

Mathematische und technische Grundlagen

- Zahlendarstellungen
- Komplement

Stellenwertsystem

TECHNISCHE UNIVERSITÄT DARMSTADT

Ein **Stellenwertsystem** ordnet jeder Ziffer anhand ihrer Position einen Stellenwert zu. Die Anordnung erfolgt von rechts nach links und jede Stelle besitzt einen konstanten Stellenwert.

Stellenwert:	10^3	10^2	10^1	10^0
Ziffern (z.B.):	5	8	3	2

höchstwertigste Stelle

niederwertigste Stelle

Um den Wert einer Dezimalzahl zu berechnen, werden die Ziffern der Zahl mit ihren Stellenwerten multipliziert und die Produkte aufaddiert.

Skript S. 127 ff

Stellenwertsysteme

TECHNISCHE UNIVERSITÄT DARMSTADT

$$a = a_{n-1}10^{n-1} + a_{n-2}10^{n-2} + \ldots + a_1 10^1 + a_0 10^0 = \sum_{i=0}^{n-1} a_i 10^i$$

$$3745 = 3 * 1000 + 7 * 100 + 4 * 10 + 5 * 1$$

Satz:
Sei $p \geq 2$ eine ganze Zahl. Dann besitzt jede ganze Zahl a mit $0 \leq a < p^n$ eine eindeutig bestimmte Darstellung der Form

$$a = \sum_{i=0}^{n-1} a_i p^i$$

mit $0 \leq a_i < p$ für alle i. Die a_i heißen Ziffern; die Darstellung nennen wir *p-adische* Darstellung von a.

Skript S. 127 ff

Explizite Angabe der Basis

TECHNISCHE UNIVERSITÄT DARMSTADT

Wo aus dem Zusammenhang nicht eindeutig hervorgeht, welche Basis die Darstellung einer Zahl verwendet, geben wir die Basis explizit als Zusatz an. Zur Darstellung der Basis verwenden wir stets das Dezimalsystem.

Beispiele:

13_{10}	=	1101_2	=	23_5	=	15_8	=	D_{16}
64_{10}	=	1000000_2	=	224_5	=	100_8	=	40_{16}
100_{10}	=	1100100_2	=	400_5	=	144_8	=	64_{16}
2505_{10}	=	100111001001_2	=	40010_5	=	4711_8	=	$9C9_{16}$

Im Hexadezimalsystem benötigen wir 16 verschiedene Ziffern, daher dienen die Buchstaben A bis F als Ziffern 10 bis 15

Skript S. 128

Gegenüberstellung von Dezimal-, Dual- und Hexadezimaldarstellung

TECHNISCHE UNIVERSITÄT DARMSTADT

Dec	Bin	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

Dec	Bin	Hex
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Komplementdarstellungen von ganzen Zahlen

Skript S. 128 f

Komplemente

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition:

Komplemente einer Zahl a in einer *p-adischen* Darstellung mit n Stellen sind Abbildungen der Menge nicht negativer ganzer $M=\{0,\ldots,p^n-1\}$ Zahlen auf sich selbst, so daß

$$k_{p-1}(a) = p^n - 1 - a$$

p = Basis
a = beliebige Zahl
n = Stellenzahl

oder

$$k_p(a) = p^n - a$$

gilt. Diese Abbildungen werden als *(p -1)-Komplement* bzw. *p-Komplement* bezeichnet.

Skript S. 128 ff

Beispiele für Komplementdarstellungen (1)

Beispiel: Komplementendarstellung Dezimalzahlen

TECHNISCHE UNIVERSITÄT DARMSTADT

Zweistellige Zahlen in Dezimalschreibweise: n = 2, p = 10

(p-1)-Komplement				*p-Komplement*			
a	$k_9(a)$	a	$k_9(a)$	a	$k_{10}(a)$	a	$k_{10}(a)$
00	99	50	49	00	00	50	50
01	98	51	48	01	99	51	49
02	97	52	47	02	98	52	48
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
48	51	98	01	48	52	98	02
49	50	99	00	49	51	99	01

Skript S. 129

Beispiele für Komplementdarstellungen (2)

Beispiel: Komplementdarstellung Dualzahlen

TECHNISCHE UNIVERSITÄT DARMSTADT

Dreistellige Zahlen in Dualschreibweise: n = 3, p = 2

(p-1)-Komplement				*p-Komplement*			
a	*k (a)*	*a*	*k (a)*	*a*	*k (a)*	*a*	*k (a)*
000	111	100	011	000	000	100	100
001	110	101	010	001	111	101	011
010	101	110	001	010	110	110	010
011	100	111	000	011	101	111	001

Skript S. 129

Nutzen der Komplementdarstellung

Komplementendarstellung

TECHNISCHE UNIVERSITÄT DARMSTADT

Definition:

Bei der K_p-*Darstellung* (p-Komplement-Darstellung) stellen wir negative Zahlen durch das Komplement ihres Betrages dar, so dass eine Folge von n Ziffern a_i eine Zahl a bedeutet nach der Regel

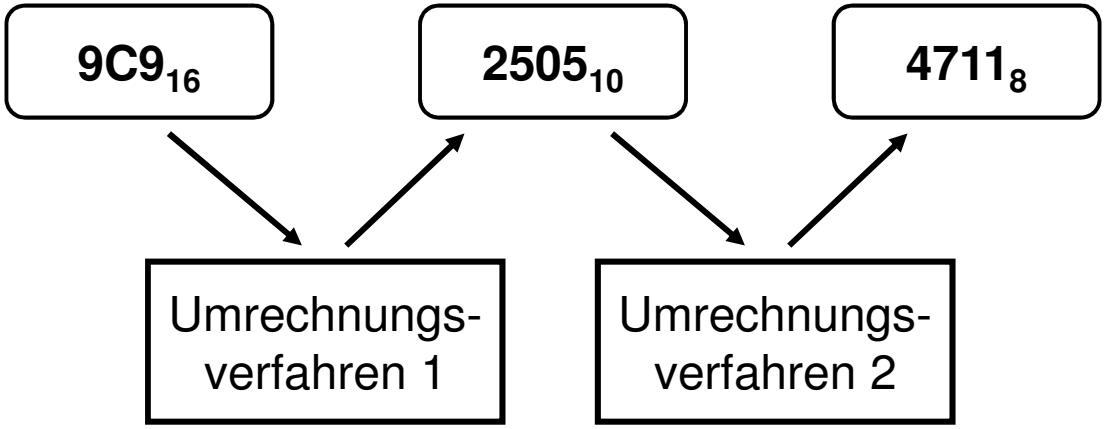
$$a = \begin{cases} \sum_{i=0}^{n-2} a_i \; p^i & wenn\ a_{n-1} = 0, \\ \sum_{i=0}^{n-2} a_i \; p^i - p^{n-1} & wenn\ a_{n-1} = p-1, \\ \text{undefiniert} & sonst \end{cases}$$

Skript S. 131

Beispiel:
Suche größte und kleinste 3-stellige Dualzahl

Umrechnungsverfahren

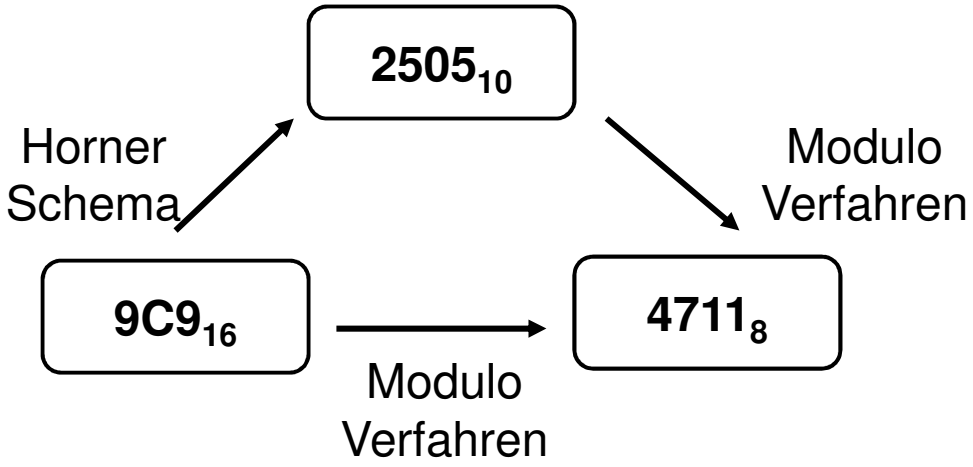
Ein Umrechnungsverfahren dient dazu, ausgehend von einer gegebenen Darstellung die Darstellung desselben Wertes mit einer anderen Basis zu ermitteln:



Skript S. 131 ff

Umrechnungsverfahren

Ein Umrechnungsverfahren dient dazu, ausgehend von einer gegebenen Darstellung die Darstellung desselben Wertes mit einer anderen Basis zu ermitteln:



Skript S. 131 ff

Hornerschema

TECHNISCHE UNIVERSITÄT DARMSTADT

Verfahren:

Zur Auswertung eines Polynoms (k-1)-ten Grades an der Stelle p beginnt man mit dem Koeffizienten a_{k-1}, der zum höchsten Exponenten p^{k-1} gehört, und bildet eine Folge von Zwischenergebnissen z_i:

$$
\begin{aligned}
z_{k-1} &= a_{k-1} \\
z_{k-2} &= pz_{k-1} + a_{k-2} \\
z_{k-3} &= pz_{k-2} + a_{k-3} \\
&\vdots \\
z_1 &= pz_2 + a_1 \\
z_0 &= pz_1 + a_0
\end{aligned}
$$

Dann ist z_0 der gesuchte Wert des Polynoms an der Stelle p.

Skript S. 132

Beispiel: Hornerschema

Beispiel zum Hornerschema

TECHNISCHE UNIVERSITÄT DARMSTADT

Berechnen des Wertes der Zahl **1457_8** nach dem Hornerschema:

mit p = 8 und k = 4 folgt:

$$z_{k-1} = z_3 := a_3 = 1$$

$$z_2 := pz_3 + a_2 = 8 \cdot 1 + 4 = 12$$

$$z_1 := pz_2 + a_1 = 8 \cdot 12 + 5 = 101$$

$$z_0 := pz_1 + a_0 = 8 \cdot 101 + 7 = 815$$

Modulo-Verfahren

TECHNISCHE UNIVERSITÄT DARMSTADT

Verfahren

Gegeben sei eine Zahl z in *k-stelliger*, *p-adischer* Darstellung. Gesucht sei die Darstellung von z zur Basis q. Man bildet eine Folge von Zwischenergebissen

$$y_0 = z \bmod q \qquad z_0 = \lfloor z/q \rfloor$$
$$y_1 = z_0 \bmod q \qquad z_1 = \lfloor z_0/q \rfloor$$
$$\vdots \qquad\qquad \vdots$$
$$y_l = z_{l-1} \bmod q \qquad z_l = \lfloor z_{l-1}/q \rfloor$$

Solange, bis $z_l = 0$ gilt. Dann sind y_i mit $j \in \{0,1,...,l\}$ die Ziffern der q-*adischen* Darstellung von z.

Skript S. 133

Beispiel: Modulo-Verfahren

Beispiel zum Modulo-Verfahren

TECHNISCHE UNIVERSITÄT DARMSTADT

Umrechnung der Zahl **815_{10}** in die Oktalschreibweise

es ist: $z = 815$, $p = 10$ und $q = 8$

$$y_0 = z \bmod q = 815 \bmod 8 = 7 \qquad z_0 = \lfloor z/q \rfloor = \lfloor 815/8 \rfloor = 101$$

$$y_1 = z_0 \bmod q = 101 \bmod 8 = 5 \qquad z_1 = \lfloor z_0/q \rfloor = \lfloor 101/8 \rfloor = 12$$

$$y_2 = z_1 \bmod q = 12 \bmod 8 = 4 \qquad z_2 = \lfloor z_1/q \rfloor = \lfloor 12/8 \rfloor = 1$$

$$y_3 = z_2 \bmod q = 1 \bmod 8 = 1 \qquad z_3 = \lfloor z_2/q \rfloor = \lfloor 1/8 \rfloor = 0$$

$$\Rightarrow \quad 1457_8$$