

# Algorithmen und Datenstrukturen in C/C++

Prof. Dr. rer. nat. Peter Gerwinski

14. Mai 2018

# Algorithmen und Datenstrukturen in C/C++

<https://gitlab.cvh-server.de/pgerwinski/ad.git>

## 1 Einführung

## 2 Einführung in C++

### 2.11 Lambda-Ausdrücke

## 3 Datenorganisation

### 3.1 Standard-Container-Templates

### 3.2 Iteratoren

### 3.3 Hash-Tabellen

### 3.4 Balancierte Bäume

### 3.5 Intelligente Zeiger

## 4 Datenkodierung

4.  $(x^2 - 1)$  Der Herr der Ringe: Manchmal ist  $1 + 1 = 0$ .

### 4.1 Fehlererkennung durch CRC

## 5 Hardwarenahe Algorithmen

## 6 Optimierung

## 7 Numerik



Änderungen  
vorbehalten

## 2 Einführung in C++

### 2.11 Lambda-Ausdrücke

- Übergabe von Funktionszeigern

```
int is_smaller (int a, int b)
{
    return a < b;
}
```

```
sort (numbers.begin (), numbers.end (), is_smaller);
```

- Stattdessen: *Lambda-Ausdrücke*

```
sort (numbers.begin (), numbers.end (), [] (int a, int b) { return a < b; });
```

- Zwischen []: *Capture-Ausdrücke*

Übergabe lokaler Variabler an die Funktion

```
sort (numbers.begin (), numbers.end (),
      [int &counter] (int a, int b) { counter++; return a < b; });
```

# 3 Datenorganisation

## 3.4 Balancierte Bäume

- **Problem:** Entartung von Bäumen  
**Lösung:** variable Anzahl von Verzweigungen
- **Problem:** Datenstruktur zu groß für Hauptspeicher  
**Lösung:** “intelligentes” Auslagern

### B-Baum:

- immer mindestens  $t$ , höchstens  $2t$  Verzweigungen
- Einfügen: wenn Knoten voll, teilen
- Löschen:
  - unterste Ebene: ggf. verschieben, ggf. verschmelzen
  - in inneren Knoten: Wert von unten ersetzen, ggf. verschmelzen
- einfachster Fall: 2-3-4-Baum

**Übungsaufgabe:** Implementieren Sie einen 2-3-4-Baum mit Methoden zum Suchen, Einfügen und Löschen

# 3 Datenorganisation

## 3.5 Intelligente Zeiger

### Warum?

- bereits freigegebene Zeiger werden u. U. weiterhin verwendet
- Speicherlecks
- uninitialisierte Zeiger
  
- `shared_ptr`
- `weak_ptr`
- `unique_ptr`
- `move()`

# 3 Datenorganisation

## 3.5 Intelligente Zeiger

### Wie?

- R-Wert-Referenztypen: `&&`
- `move()`-Funktion

### Literatur:

- [http://thbecker.net/articles/rvalue\\_references/section\\_01.html](http://thbecker.net/articles/rvalue_references/section_01.html)
- <http://www.artima.com/cppsource/rvalue.html>

## 4 Datenkodierung

**4.  $(x^2 - 1)$  Der Herr der Ringe: Manchmal ist  $1 + 1 = 0$ .**

**4.  $(x^2 - 1).x$  Motivation**

Man kann auch mit sehr merkwürdigen Objekten wie mit „ganz normalen“ Zahlen rechnen.

Anwendungen:

- Funktionsweise von Computern ( $\longrightarrow$  Rechnertechnik)
- Fehlererkennung
- Fehlerkorrektur
- Verschlüsselung
- Digitale Signaturen

## 4 Datenkodierung

4.  $(x^2 - 1)$  Der Herr der Ringe: Manchmal ist  $1 + 1 = 0$ .

4.  $(x^2 - 1) \cdot (x + 1)$  Gruppen

**Definition:** Sei  $G$  eine Menge,  $*$  eine Verknüpfung auf  $G$ . Wenn

- $\forall a, b, c \in G: (a * b) * c = a * (b * c)$  (Assoziativgesetz),
- $\exists e \in G: \forall a \in G: a * e = e * a = a$  (neutrales Element),
- $\forall a \in G: \exists a^{-1} \in G: a * a^{-1} = a^{-1} * a = e$  (inverses Element),

dann heißt  $(G, *)$  eine *Gruppe*.



## 4 Datenkodierung

4.  $(x^2 - 1)$  Der Herr der Ringe: Manchmal ist  $1 + 1 = 0$ .

4.  $(x^2 - 1) \cdot (x + 1)$  Gruppen

**Definition:** Sei  $G$  eine Menge,  $*$  eine Verknüpfung auf  $G$ . Wenn

- $\forall a, b, c \in G: (a * b) * c = a * (b * c)$  (Assoziativgesetz),
- $\exists e \in G: \forall a \in G: a * e = e * a = a$  (neutrales Element),
- $\forall a \in G: \exists a^{-1} \in G: a * a^{-1} = a^{-1} * a = e$  (inverses Element),

dann heißt  $(G, *)$  eine *Gruppe*.

**Definition:** Sei  $(G, *)$  eine Gruppe. Wenn zusätzlich

- $\forall a, b \in G: a * b = b * a$  (Kommutativgesetz),

dann heißt  $(G, *)$  eine *kommutative Gruppe*.

## 4. $(x^2 - 1)$ Der Herr der Ringe: Manchmal ist $1 + 1 = 0$ .

### 4. $(x^2 - 1) \cdot (x + 2)$ Ringe

**Definition:** Sei  $R$  eine Menge; seien  $+$  und  $\cdot$  Verknüpfungen auf  $R$ . Wenn

- $(R, +)$  eine kommutative Gruppe ist,
- $\forall a, b, c \in R: (a \cdot b) \cdot c = a \cdot (b \cdot c)$  (Assoziativgesetz),
- $\forall a, b, c \in R: (a + b) \cdot c = a \cdot c + b \cdot c$  und  $a \cdot (b + c) = a \cdot b + a \cdot c$  (Distributivgesetze),

dann heißt  $(R, +, \cdot)$  ein *Ring*.

## 4. $(x^2 - 1)$ Der Herr der Ringe: Manchmal ist $1 + 1 = 0$ .

### 4. $(x^2 - 1) \cdot (x + 2)$ Ringe

**Definition:** Sei  $R$  eine Menge; seien  $+$  und  $\cdot$  Verknüpfungen auf  $R$ . Wenn

- $(R, +)$  eine kommutative Gruppe ist,
- $\forall a, b, c \in R: (a \cdot b) \cdot c = a \cdot (b \cdot c)$  (Assoziativgesetz),
- $\forall a, b, c \in R: (a + b) \cdot c = a \cdot c + b \cdot c$  und  $a \cdot (b + c) = a \cdot b + a \cdot c$  (Distributivgesetze),

dann heit  $(R, +, \cdot)$  ein *Ring*.

**Definition:** Sei  $(R, +, \cdot)$  ein Ring. Wenn zustzlich

- $\forall a, b \in R: a \cdot b = b \cdot a$  (Kommutativgesetz),

dann heit  $(R, +, \cdot)$  ein *kommutativer Ring*.

## 4. $(x^2 - 1)$ Der Herr der Ringe: Manchmal ist $1 + 1 = 0$ .

### 4. $(x^2 - 1) \cdot (x + 2)$ Ringe

**Definition:** Sei  $R$  eine Menge; seien  $+$  und  $\cdot$  Verknüpfungen auf  $R$ . Wenn

- $(R, +)$  eine kommutative Gruppe ist,
- $\forall a, b, c \in R: (a \cdot b) \cdot c = a \cdot (b \cdot c)$  (Assoziativgesetz),
- $\forall a, b, c \in R: (a + b) \cdot c = a \cdot c + b \cdot c$  und  $a \cdot (b + c) = a \cdot b + a \cdot c$  (Distributivgesetze),

dann heißt  $(R, +, \cdot)$  ein *Ring*.

**Definition:** Sei  $(R, +, \cdot)$  ein Ring. Wenn zusätzlich

- $\forall a, b \in R: a \cdot b = b \cdot a$  (Kommutativgesetz),

dann heißt  $(R, +, \cdot)$  ein *kommutativer Ring*.

**Definition:** Sei  $(R, +, \cdot)$  ein (kommutativer) Ring. Wenn zusätzlich

- ein  $e \in R$  existiert, so daß für alle  $a \in R$  gilt:  $a \cdot e = e \cdot a = a$  (neutrales Element),

dann heißt  $(R, +, \cdot)$  ein *(kommutativer) Ring mit 1*.

## 4. $(x^2 - 1)$ Der Herr der Ringe: Manchmal ist $1 + 1 = 0$ .

## 4. $(x^2 - 1).(x + 3)$ Körper

**Definition:** Sei  $K$  eine Menge; seien  $+$  und  $\cdot$  Verknüpfungen auf  $K$ . Wenn

- $(K, +, \cdot)$  ein Ring mit 1 ist und
- $(K \setminus \{0\}, \cdot)$  eine kommutative Gruppe ist,

dann heißt  $(K, +, \cdot)$  ein *Körper*.

## 4.1 Fehlererkennung durch CRC

- Verallgemeinerung von „gerade/ungerade“:  
Rest bei Polynomdivision über Körper mit 2 Elementen ( $1 + 1 = 0$ )
- Rest = Prüfwert

1 0 1 1 0 1 0 0 1 1 1 0 0 0 : 1 1 0 1  
1 1 0 1

1 1 0 0  
1 1 0 1

Prüfwert ermitteln

1 1 0 0  
1 1 0 1

1 1 1 1  
1 1 0 1

1 0 0 0  
1 1 0 1

1 0 1 0  
1 1 0 1

1 1 1

## 4.1 Fehlererkennung durch CRC

- Verallgemeinerung von „gerade/ungerade“:  
Rest bei Polynomdivision über Körper mit 2 Elementen ( $1 + 1 = 0$ )
- Rest = Prüfwert

1 0 1 1 0 1 0 0 1 1 1    1 1 1    :    1 1 0 1  
1 1 0 1

1 1 0  
1 1 0 1

Daten prüfen

1 1 0 0  
1 1 0 1

1 1 1 1  
1 1 0 1

1 0 1 1  
1 1 0 1

1 1 0 1  
1 1 0 1

0

# Algorithmen und Datenstrukturen in C/C++

<https://gitlab.cvh-server.de/pgerwinski/ad.git>

## 1 Einführung

## 2 Einführung in C++

## 3 Datenorganisation

### 3.1 Standard-Container-Templates

### 3.2 Iteratoren

### 3.3 Hash-Tabellen

### 3.4 Balancierte Bäume

### 3.5 Intelligente Zeiger

## 4 Datenkodierung

### 4.( $x^2 - 1$ ) Mathematische Grundlagen

### 4.1 Fehlererkennung durch CRC

## 5 Hardwarenahe Algorithmen

## 6 Optimierung

## 7 Numerik



Änderungen  
vorbehalten