

Julius-Maximilians-Universität Würzburg  
Institut für Informatik  
Lehrstuhl für Informatik I  
Effiziente Algorithmen und wissensbasierte Systeme

## **Bachelorarbeit**

# **Titel der Arbeit**

Testvorname Testnachname

Eingereicht am XX. YY 20ZZ

Betreuer:

Prof. Dr. Alexander Wolff  
Dipl.-Inf. Max Mustermann

# Inhaltsverzeichnis

<b>1</b>	<b>Kapitel sind ganz einfach</b>	<b>3</b>
1.1	Abschnitte ebenfalls . . . . .	3
1.2	Zweiter Abschnitt . . . . .	3
1.2.1	Ein Unterabschnitt . . . . .	4
1.2.2	Noch ein Unterabschnitt . . . . .	5

# 1 Kapitel sind ganz einfach

## 1.1 Abschnitte ebenfalls

Hier fängt der Text an.

**Satz 1.1** (Finkscher Hauptsatz). *Wichtige und grundlegende Sätze lassen sich leicht hervorheben.*

*Beweis.* Der Satz gilt offensichtlich, denn

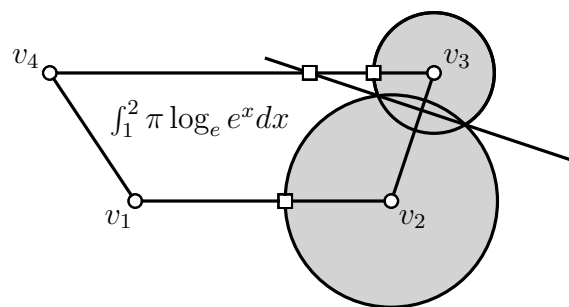
$$\sum_{i=1}^n 1 = n$$

Zudem wird der Beweis automatisch mit einem q.e.d.-Symbol beendet. □

Auf Sätze, wie z.B. Satz 1.1, lässt sich mithilfe des Befehls `\ref{labelname}` verweisen, wenn man in der Satz-Umgebung einen „Label“ mit `\label{labelname}` gesetzt hat. Genauso können wir auf den nächsten Abschnitt, also Abschnitt 1.2, verweisen. Üblicherweise beginnt man einen Labelnamen mit dem Typ der Umgebung, auf die man verweist, also z.B. `\label{fig:trapez}` für eine Abbildung (engl. *figure*). Ach ja, zum Hervorheben (engl. *emphasize*) eines *neuen Begriffs* verwendet man den Befehl `\emph{neuer Begriff}`, wenn der neue Begriff zum ersten Mal verwendet wird.

## 1.2 Zweiter Abschnitt

**Definition 1.2.** Definitionen lassen sich leicht erstellen.



**Abb. 1.1:** Das ist eine Abbildung.

Auch Abbildungen, wie z.B. Abbildung 1.1, sind schnell eingefügt. Im Allgemeinen braucht man die Endung der Bilddatei beim Einbinden mit `\includegraphics` nicht mit anzugeben.

### 1.2.1 Ein Unterabschnitt

Zu viele Unterebenen nach Möglichkeit vermeiden. Wir wollen hier nur zeigen, dass es mit der `algorithm`-Umgebung (aus dem Paket `algorithm2e.sty`) nicht schwer ist Algorithmen in Pseudocode zu setzen, siehe Algorithmus 1.

---

**Algorithmus 1:** BinäreSuche(Feld  $A$ , ganze Zahl  $n$ , Element  $x$ )

---

**Eingabe :** sortiertes Feld  $A$ , Länge  $n$ , gesuchtes Element  $x$   
**Ausgabe :** `true` genau dann, wenn  $x$  in  $A$  enthalten ist

```
1  $l = 0$ 
2  $r = n - 1$ 
3 while  $l \leq r$  do
4    $m = \lfloor (l + r)/2 \rfloor$ 
5   if  $A[m] == x$  then
6     return true
7   else if  $x < A[m]$  then
8      $r = m - 1$ 
9   else
10     $l = m + 1$ 
11 return false
```

---

Das gleiche geht problemlos auch ohne Zeilennummern, siehe Algorithmus 2. Dazu benutzt man einfach in der `algorithm`-Umgebung den Befehl `\LinesNotNumbered`.

---

**Algorithmus 2:** BinäreSucheOhneZeilennum(Feld  $A$ , ganze Zahl  $n$ , Element  $x$ )

---

**Eingabe :** sortiertes Feld  $A$ , Länge  $n$ , gesuchtes Element  $x$   
**Ausgabe :** `true` genau dann, wenn  $x$  in  $A$  enthalten ist

```
 $l = 0$ 
 $r = n - 1$ 
while  $l \leq r$  do
   $m = \lfloor (l + r)/2 \rfloor$ 
  if  $A[m] == x$  then
    return true
  else if  $x < A[m]$  then
     $r = m - 1$ 
  else
     $l = m + 1$ 
return false
```

---

### **1.2.2 Noch ein Unterabschnitt**

Auch Verweise auf ältere Resultate, wie das von Mustermann und Musterfrau [MM11], sind ganz einfach.

# Literaturverzeichnis

- [MM11] Max Mustermann und Monika Musterfrau: Beispiele in der Anwendung. *Beispiele und Muster*, 61(2):306–320, 2011.