



---

---

## Titel der Arbeit

---

---

Zur Erlangung des akademischen Grades  
*Bachelor of Science*  
im Studiengang Informatik

*Vorgelegt von:*

**Vorname Nachname**

Matrikelnummer: 123456

mail@wwu.de

*Betreuer:*

**Prof. Dr. Max Mustermann**

max.mustermann@wwu.de

**Jane Doe**

jane.doe@wwu.de

Erstgutachter:

Prof. Dr. Max Mustermann, Institut für Informatik

Zweitgutachter:

Prof. Dr. John Doe, Institut für Informatik

Münster, 13. Januar 2020



## Vorwort

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



## **Inhaltsverzeichnis**

<b>1</b>	<b>Beispiele</b>	<b>1</b>
1.1	Algorithmen . . . . .	1
1.2	Code-Beispiele . . . . .	1
1.3	Grafiken . . . . .	3
1.4	Theoreme, Sätze, Lemmata, Definitionen . . . . .	3
1.5	Grafiken . . . . .	3
1.6	Einige nützliche Tools . . . . .	3
<b>2</b>	<b>Zitationen</b>	<b>5</b>
	<b>Literatur</b>	<b>7</b>



# 1 Beispiele

Nachfolgend zeige ich einige kurze Ausschnitte meiner Bachelorarbeit, um beispielhaft zu zeigen, wie bestimmte Umgebungen genutzt werden können. Man kann in allen Fällen noch viel mehr machen, aber so entsteht vielleicht eine Idee.

## 1.1 Algorithmen

Ein wichtiger Aspekt ist oft Pseudocode. Hierzu gibt es eine Vielzahl von packages, ich selbst nutze `algorithmicx` in der `algpseudocode`-Variante. Die Bedienung ist recht intuitiv. Nachfolgend ein Beispiel zur diskreten Fréchet-Distanz auf Trajektorien (zeitlich geordneten Punktfolgen).

---

**Algorithmus 1.1** zur Berechnung der diskreten Fréchet-Distanz zwischen zwei Trajektorien  $A = (a_1, \dots, a_n)$  und  $B = (b_1, \dots, b_m)$  mit  $1 \leq n, m$ .

---

**Require:** Es gilt  $|A|, |B| \geq 1$ .

```
1: function DISCRETEFRÉCHET(Trajektorie  $A$ , Trajektorie  $B$ )
2:   Erzeuge Feld  $D[1, \dots, n][1, \dots, m]$ 
3:   for  $i := 1$  to  $n$  do
4:     for  $j := 1$  to  $m$  do
5:       if  $i = j = 1$  then
6:          $D[1][1] := d(a_1, b_1)$  // Basisfall
7:       else if  $i = 1$  then // zweite Bedingung
8:          $D[i][j] := \max\{d(a_i, b_j), D[i][j-1]\}$ 
9:       else if  $j = 1$  then // dritte Bedingung
10:         $D[i][j] := \max\{d(a_i, b_j), D[i-1][j]\}$ 
11:       else //  $i, j > 1$ 
12:         $D[i][j] := \max\{d(a_i, b_j), \min\{D[i][j-1], D[i-1][j], D[i-1][j-1]\}\}$ 
13:   return  $D[n][m]$ 
```

---

## 1.2 Code-Beispiele

Für Code-Beispiele nutze ich `minted`, in der config-Datei habe ich Makros für Java, C und C++ hinzugefügt. Das geht entweder über ausgelagerte Dateien oder über direktes Einfügen des Codes. Labels werden automatisch mit dem Prefix "list:" versehen. Dieser Code liefert die nachfolgende Ausgabe:

---

```
1 \begin{JavaCode}{Dies ist eine Beschreibung.}{label}
2   // Simple implementation of recursive equation for coupling distance.
3   for (int i = 0; i < pointsA.length; i++) {
4     for (int j = 0; j < pointsB.length; j++) {
5       double deltaCurrent =
6       delta.getDistance(pointsA[i].getxValue(),
        ↪ pointsA[i].getyValue(), pointsB[j].getxValue(),
        ↪ pointsB[j].getyValue());
```

## 1 BEISPIELE

```
7         if (j == 0 && i == 0) {
8             resultMatrix[i][j] = deltaCurrent;
9         } else if (i == 0) {
10            resultMatrix[i][j] = distanceOperator(deltaCurrent,
11            ↪ resultMatrix[i][j-1]);
12        } else if (j == 0) {
13            resultMatrix[i][j] = distanceOperator(deltaCurrent,
14            ↪ resultMatrix[i-1][j]);
15        } else {
16            resultMatrix[i][j] =
17            distanceOperator(deltaCurrent, getMin(resultMatrix[i-1][j],
18            ↪ resultMatrix[i][j-1], resultMatrix[i-1][j-1]));
19        }
20    }
21 }
22 \end{JavaCode}
```

---

```
5 // Simple implementation of recursive equation for coupling distance.
6 for (int i = 0; i < pointsA.length; i++) {
7     for (int j = 0; j < pointsB.length; j++) {
8         double deltaCurrent =
9         delta.getDistance(pointsA[i].getXValue(), pointsA[i].getYValue(),
10        ↪ pointsB[j].getXValue(), pointsB[j].getYValue());
11        if (j == 0 && i == 0) {
12            resultMatrix[i][j] = deltaCurrent;
13        } else if (i == 0) {
14            resultMatrix[i][j] = distanceOperator(deltaCurrent,
15            ↪ resultMatrix[i][j-1]);
16        } else if (j == 0) {
17            resultMatrix[i][j] = distanceOperator(deltaCurrent,
18            ↪ resultMatrix[i-1][j]);
19        } else {
20            resultMatrix[i][j] =
21            distanceOperator(deltaCurrent, getMin(resultMatrix[i-1][j],
22            ↪ resultMatrix[i][j-1], resultMatrix[i-1][j-1]));
23        }
24    }
25 }
```

---

Listing 1.1: Dies ist eine Beschreibung.

Man kann dabei die Zeilennummern mit dem optionalen Parameter „linenos=false“ ausstellen oder mit „firstline=151“ bzw. „lastline=160“ manuell einschränken, welchen Bereich des Code-Ausschnittes/der Datei man zeigen möchte (bietet sich vor allem beim Einbinden von Dateien an). Mit dem Makro `\code` könnt ihr den Startwert der Zeilennummern setzen. **Wenn ihr das außerhalb einer Gruppe/Umgebung**



setzt, ist die Änderung global. Also am besten den Befehl und den COde, auf den es sich auswirken soll, mit geschweiften Klammern versehen :)

Das Einbinden von Code aus Dateien funktioniert analog mit

---

```
\inputJava[parameters]{filepath}{caption}{label}
```

---

Dabei sind die Parameter wieder optional. Die gleichen Makros sind auch für C und C++ angelegt (ersetzt „Java“ durch „C“ bzw. „C++“).

### 1.3 Grafiken

### 1.4 Theoreme, Sätze, Lemmata, Definitionen

### 1.5 Grafiken

### 1.6 Einige nützliche Tools

## 1 BEISPIELE

Dies ist ein Beispiel, wie man einfach zitieren kann [1], natürlich funktionieren auch bestimmte Kapitel oder Abschnitte [2, Kapitel 1].

## 2 Zitationen

Die ist ein Beispiel, wie man zitiert[1]. Natürlich geht das auch mit Angabe des Kapitels[2, Kapitel 1].



## Literatur

- [1] John Smith und Joan Smith. „A very interesting paper“. In: *A very good journal* (2009).
- [2] John Smith et al. *A very interesting book*. 2010.



## Eidesstattliche Erklärung

Hiermit versichere ich, dass die vorliegende Arbeit über „*Titel der Arbeit*“ selbstständig verfasst worden ist, dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt worden sind und dass die Stellen der Arbeit, die anderen Werken – auch elektronischen Medien – dem Wortlaut oder Sinn nach entnommen wurden, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht worden sind.

---

Vorname Nachname, Münster, 13. Januar 2020

Ich erkläre mich mit einem Abgleich der Arbeit mit anderen Texten zwecks Auffindung von Übereinstimmungen sowie mit einer zu diesem Zweck vorzunehmenden Speicherung der Arbeit in eine Datenbank einverstanden.

---

Vorname Nachname, Münster, 13. Januar 2020