



Fakultät für Mathematik

Lehrstuhl für Angewandte Geometrie und Diskrete Mathematik

# **Adaption eines didaktischen Konzepts zur Darstellung weiterführender Graphalgorithmen in einer Web-Applikation**

Chinese Postman Problem

Algorithmus von Floyd-Warshall

Algorithmus von Hierholzer

Algorithmus von Hopcroft und Karp

Ungarische Methode

**Abschlussbericht für ein interdisziplinäres Projekt von  
Mark-Johannes Becker, Aleksejs Voroncovs, Ruslan Zabrodin**

Themensteller: Prof. Dr. Peter Gritzmann

Betreuer: M.Sc. Wolfgang Ferdinand Riedl

Abgabedatum: 14. Mai 2015



Hiermit erkläre ich, dass ich diese Arbeit selbstständig angefertigt und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 14. Mai 2015

---

Mark-Johannes Becker, Aleksejs Voroncovs, Ruslan Zabrodin



## **Abstract**

Here we give a short summary of the project or thesis of length at most a quarter of a page.

## **Zusammenfassung**

Hier schreibt man eine kurze Zusammenfassung der Arbeit im Umfang von maximal einer Viertelseite.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Aufbau der Webanwendungen</b>	<b>3</b>
2.1	Algorithmus von Floyd-Warshall . . . . .	3
2.2	Algorithmus von Hierholzer (Mark) . . . . .	3
2.3	Algorithmus von Hopcroft und Karp . . . . .	3
2.4	Ungarische Methode . . . . .	3
2.5	Chinese Postman Problem . . . . .	3
<b>3</b>	<b>Implementierung</b>	<b>5</b>
3.1	Installation (Mark) . . . . .	5
3.2	MathJAX (Mark) . . . . .	5
3.3	Bipartite Graphen . . . . .	7
3.4	Multigraphen . . . . .	7
3.5	Zufällig generierte Fragen (Mark) . . . . .	7
3.6	Gemeinsam genutzte Dateien (Mark) . . . . .	8
<b>4</b>	<b>Zusammenfassung</b>	<b>9</b>
	<b>Abbildungsverzeichnis</b>	<b>11</b>
	<b>Index</b>	<b>11</b>
	<b>Literatur</b>	<b>13</b>





# Kapitel 1

## Einleitung

Einleitung Motivation Referenz auf führende IDPs (Quellen!) didaktisches Konzept, kurz  
Aufbau der Dokumentation, inkl. Benennung wer hat was gemacht (Benotung)



# Kapitel 2

## Aufbau der Webanwendungen

funktionale Beschreibung der Apps jedes Tab

### 2.1 Algorithmus von Floyd-Warshall

Besonderheiten Visualisierung Forschungsfragen

### 2.2 Algorithmus von Hierholzer (Mark)

Besonderheiten Visualisierung Forschungsfragen

### 2.3 Algorithmus von Hopcroft und Karp

Besonderheiten Visualisierung Forschungsfragen

### 2.4 Ungarische Methode

Besonderheiten Visualisierung Forschungsfragen

### 2.5 Chinese Postman Problem

Besonderheiten Visualisierung Forschungsfragen



# Kapitel 3

## Implementierung

### 3.1 Installation (Mark)

Zur Installation sind keine speziellen Anforderungen zu erfüllen. Die Webapplikationen wurden vollständig mittels HTML5, JavaScript und CSS implementiert, sodass keine zusätzliche serverseitige Software benötigt wird. Zum Aufrufen der Anwendungen wird ein moderner Webbrowser benötigt.

Die Bereitstellung erfolgt über den Online Dienst GitHub, der das verteilte Versionskontrollsystem Git benutzt. Alle Anwendungen liegen in einem gemeinsamen Repository, welches unter <https://github.com/herzog31/adv-graph-algorithms> erreichbar ist. Zur Installation kann man entweder auf der Repository Seite die letzte freigegebene Version (Release) als Zip oder Tar Archiv herunterladen oder die aktuellste Version des Repositories mittels folgendem Befehl in das aktuelle Verzeichnis kopieren.

```
1 git clone -b master https://github.com/herzog31/adv-graph-algorithms ↵  
   ↪ algorithms.git
```

**Abbildung 3.1:** Befehl zum Kopieren des GitHub Repositories

Um die Anwendungen als Webanwendungen online bereitzustellen wird ein Webserver benötigt. Hier empfiehlt sich die Installation des Apache oder nginx HTTP Servers.

Die lokale Ausführung ist ohne einen Webserver möglich. Verschiedene Webbrowser besitzen allerdings eine Sicherheitsrichtlinie, die das Öffnen von lokalen Dateien über JavaScript verbieten. Diese Sicherheitsrichtlinie kann jedoch durch spezielle Einstellungen umgangen werden. Für Google Chrome ist dies über den Start Parameter `-allow-file-access-from-files` möglich.

### 3.2 MathJAX (Mark)

Unter den Tabs "Beschreibung des Algorithmus" werden die komplizierte Algorithmen, wie bspw. die Ungarische Methode, in möglichst einfachen Worten als Fließtext erklärt. Wir gehen davon aus, dass der Nutzer während der Bearbeitung der Forschungsaufgaben,

Dazu haben wir zusätzlich zur textuellen Beschreibung, wichtige Punkte der Algorithmen auch als mathematische Formeln dargestellt. Der Nutzer erlangt so durch das erste Lesen der Beschreibung ein Grundverständnis der Algorithmen und kann während der Bearbeitung aufkommende Fragen durch die dargestellten Formeln schneller erschließen.

Damit die Bibliothek in den Webanwendungen genutzt werden kann, muss sie wie folgt eingebunden werden (vgl. Abbildung 3.2.

**Abbildung 3.2:** Einbinden der MathJAX Bibliothek

### Abbildung 3.3: MathJAX Beispiel: HTML Code

<sup>1</sup><https://www.mathjax.org/>

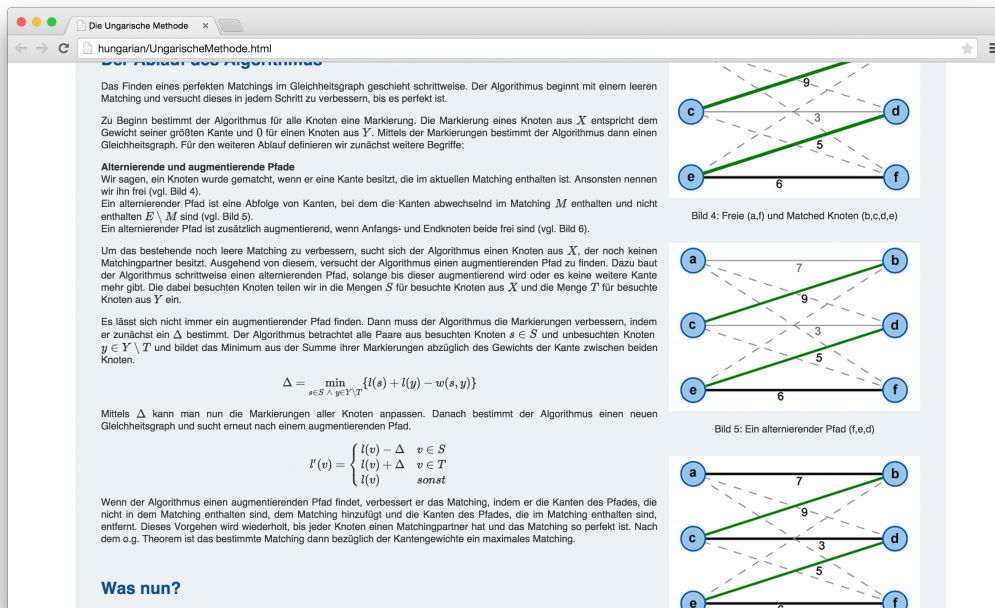


Abbildung 3.4: MathJAX Beispiel: Darstellung im Browser

```
1 MathJax.Hub.Queue(["Typeset", MathJax.Hub]);
```

Abbildung 3.5: Befehl zum erneuten Rendern von Ausdrücken im HTML Dokument

### 3.3 Bipartite Graphen

Motivation Beispiel

### 3.4 Multigraphen

Motivation Beispiel

### 3.5 Zufällig generierte Fragen (Mark)

Motivation Beispiel

## **3.6 Gemeinsam genutzte Dateien (Mark)**

Motivation Beispiel



## **Kapitel 4**

### **Zusammenfassung**

Zusammenfassung der wesentlichen Punkte



# Abbildungsverzeichnis

3.1	Repository Kopieren . . . . .	5
3.2	MathJAX Einbindung . . . . .	6
3.3	MathJAX Beispiel . . . . .	6
3.4	MathJAX Beispiel . . . . .	7
3.5	MathJAX Rendern . . . . .	7

Alle Abbildungen in diesem Dokument stammen vom Autor selbst. Zur Erzeugung wurde das großartige  $\text{\TeX}$ -Paket TikZ von Till Tantau verwendet, vgl. [[Tan08](#)].



# Literatur

- [Tan08] T. Tantau. *The TikZ and PGF Packages. Manual for version 2.00*. Lübeck, Feb. 2008.