Fachhochschule Köln
Cologne University of Applied Sciences

Prof. Dr. Heiner Klocke
Fakultät für Informatik und
Ingenieurwissenschaften

Algorithmik-Praktikum

WS 2010/2011

Thema: Fibonacci-Heaps applied to pathfinding and spanning tree algorithms

A. Organisatorisches

Das Algorithmik-Praktikum im WS 2010/2011 wird als Teamprojekt durchgeführt. Bilden Sie dazu Teams mit 2 oder 3 Teilnehmern und melden sich online unter dem Link auf meiner Homepage an.

Erasmus students! If you think that this project in algorithms is too comprehensive for finishing it within your stay here in Gummersbach, please contact me. I can define another convenient exercise that you can finish within a appropriate time. Don't hesitate to contact me.

1. Termine

Es wird **5 Pflichttermine** geben:

1. Vorstellung des Projekts und Klärung aller organisatorischen Fragen
2. Meilenstein zur theoretischen und experimentellen **Analyse.** Vorstellung der Testumgebung.
3. Meilenstein zu **Fibonacci-Heaps**
4. Meilenstein zu **Path-Finding-** und **Tree-Spanning-Algorithmen** mit **Fibonacci-Heaps**. **Analysen**!
5. Präsentation des Projekts mit Vortrag und Vorführung der Software

2. Dokumentation im Projekt-Wiki

Jedes Team muss begleitend zu den Meilensteinen den Fortschritt des Projekts im Wiki

http://www.software-quality.fh-koeln.de/algorithmik/index.php/Gruppen_WS1011

dokumentieren. Die lauffähigen Softwarelösungen und die Projekt-Dokumentation sind im Algorithmik-Wiki unter Ihrem Teamnamen zu speichern. Die **Termine** für die Meilensteine finden Sie auf meiner Homepage. Zwei Tage vor einer Präsentation muss die für den Meilenstein aktuelle Dokumentation im Wiki stehen.

3. Bonuspunkte

Das Algorithmik-Projekt wird mit bis zu **36 Punkten** (9 pro Termin) bewertet, die auf die Klausurpunktzahl angerechnet werden. Die erreichten Punkte gelten für die auf die Vorlesung drei folgenden Klausurtermine.

4. Meilensteinpräsentationen und Dokumentation

Die Präsentation der Ergebnisse zu den jeweiligen Meilensteinen soll als **Folienpräsentation** im PDF-Format vorbereitet und **2 Tage vorher im Algorithmik-Wiki** gespeichert werden. Verwenden Sie dazu den **Dateinamen „<Teamname>_MS_<Meilensteinnummer>**“. Eine **Meilensteinpräsentation** soll pro Team ca. **15 Minuten** dauern.

4.1 Präsentation

1. Darstellung und Erklärung der Meilensteinaufgabe
2. Präsentation des Lösungskonzepts: Ihre Ideen, Methoden, Vorgehensweisen und Lösungen
3. Vorführung der für den Meilenstein entwickelten Programme. Bitte keinen Quellcode zeigen.
4. Diskussion.
5. Änderungen und Verbesserungen festlegen, die bis zum nächsten Meilenstein oder gegebenenfalls bis zu einem zusätzlichen Termin durchzuführen sind.
6. Vorlage und Besprechung der Dokumentation

4.2 Dokumentation

Betrachten Sie die Dokumentation als ein wichtiges Projektdokument Ihres Studiums. Gestalten Sie das Dokument sprachlich, inhaltlich und strukturell so, dass Sie es bei einer Bewerbung vorgelegen können. Schicken Sie 2 Tage vor dem Meilensteintermin die Dokumentation im PDF-Format an alex.maier@fh-koeln.de und heinrich.klocke@fh-koeln.de mit dem **Betreff „<Teamname>_MS_<Meilensteinnummer>“.** Struktur der Dokumentation:

1. **Titelseite**. Ein Muster steht auf meiner Homepage.
2. Genaues **Inhaltsverzeichnis**
3. Detaillierte **Aufgabenstellung**
4. Sorgfältige Dokumentation der **Meilensteine**
    - Darstellung der **Teilaufgabe**
    - Erläuterung der verwendeten algorithmischen **Konzepte** und **Lösungsideen**
    - Beschreibung der **realisierten Teilaufgabe**. Erklärung der Programmfunktionalität z.B. auch mit kommentierten Screenshots vom User Interface
    - **Überprüfung** der sprachlichen und fachlichen Korrektheit sowie der Rechtschreibung
5. **Literaturliste**: Bücher, Artikel und Internetquellen

B. Projektbeschreibung

1. Thema

Path-Finding- und **Tree-Spanning-Algorithmen** unter Verwendung von **Priority-Queues** und deren **Analysen**

Jedes Team kann zwischen einer Path-Finding und einer Tree-Spanning-Anwendung wählen.

2. Lernziele:

- Theoretische und experimentelle Analyse von Algorithmen
- Einsatz von Priority-Queues als wichtige Hilfsdatenstruktur für Graphalgorithmen zur Verbesserung der Laufzeitperformance
- Anwendung von Graphalgorithmen für praktische Anwendungen

3. Worauf kommt es an?

Alle für dieses Projekt Themen werden in der Vorlesung ausführlich behandelt und zwar in der Reihenfolge Analyse, Priority-Queues und Graphalgorithmen. Die Graphalgorithmen selbst sind nicht schwierig zu verstehen. Wesentlich anspruchsvoller und komplexer sind die Analyseverfahren und die Priority-Queues, insbesondere die Fibonacci-Heaps. Es ist wichtig, dass Sie genau verstehen, wie Fibonacci-Heaps arbeiten und welche Performance durch sie gewonnen wird. In dem Projekt kommt es somit besonders auf die effiziente Anwendung von Hilfsdatenstrukturen, die experimentellen Analysen sowie die Darstellung der Ergebnisse an.

4. Experimentelle Analyse

Entwickeln Sie für die experimentelle Algorithmenanalyse eine Testumgebung, mit der Sie den Powertest für beliebige Polynomzeitalgorithmen anwenden können (Kapitel 1.2.2 der Vorlesung). Verwenden Sie zum Testen dieser Umgebung einen $O(n^2)$-Algorithmus, z.B. BubbleSort. Die Tests müssen genau geplant (Testdaten auswählen bzw. erzeugen!), sorgfältig durchgeführt (korrekt messen!) und genau protokolliert werden. Stellen Sie die (x, y)-Paare und die logarithmisch transformierten (x', y')-Paare grafisch dar. Verwenden dazu Sie z.B. die Bibliothek **Ptplot 5.8** Bibliothek (http://ptolemy.eecs.berkeley.edu/java/ptplot/). Berechnen Sie für jede (x', y')-Tabelle den Korrelationskoeffizienten, um die Qualität der Daten zu prüfen.

Die Testumgebung soll beim **ersten Meilensteintermin** vorgestellt werden.

5. Path-Finding Aufgabenbeschreibung

- Entwickeln Sie eine Software, mit der die/eine kürzeste oder schnellste Route zwischen zwei Orten berechnet werden kann. Wählen Sie dazu einen Graph-Algorithmus aus der Vorlesung, der für diese Aufgabe am besten geeignet ist. Zur Verwaltung der Daten müssen Fibonacci-Heaps als Hilfsdatenstruktur verwendet werden.

- Für die Routenplanung ist eine GUI zu entwickeln, über die alle Optionen der Route wie Start, Ziel, schnellste oder kürzeste Route, etc. eingegeben werden können.
- Die Ergebnisse der Routenplanung sind grafisch und numerisch darzustellen. Ebenso soll der Verlauf des Algorithmus anhand einer geeigneten grafischen Darstellung beobachtbar sein.
- Parallel zum Ablauf des Path-Finding-Algorithmus soll ein Monitoring für die Priority-Queue-Struktur durchgeführt werden. Überlegen Sie sich, welche Informationen dazu sinnvoll und notwendig sind und wie diese dargestellt werden.
- Testen Sie die Path-Finding-Anwendung zunächst für kleine ungerichtete Graphen. Testen Sie danach mit realen Daten. Verwenden Sie dazu Straßennetze, die als Dateien unter ***http://www.cs.fsu.edu/~lifeifei/SpatialDataset.htm*** heruntergeladen werden können (z.B. Bild 1). Führen Sie für diese Daten experimentelle Analysen durch. Jedes Analyse-Experiment ist sorgfältig zu planen, durchzuführen und zu dokumentieren. Was wird analysiert? Welche Daten sind dafür notwendig? Wie wird der Test durchgeführt und protokolliert? Wie werden die Ergebnisse numerisch und grafisch dargestellt?

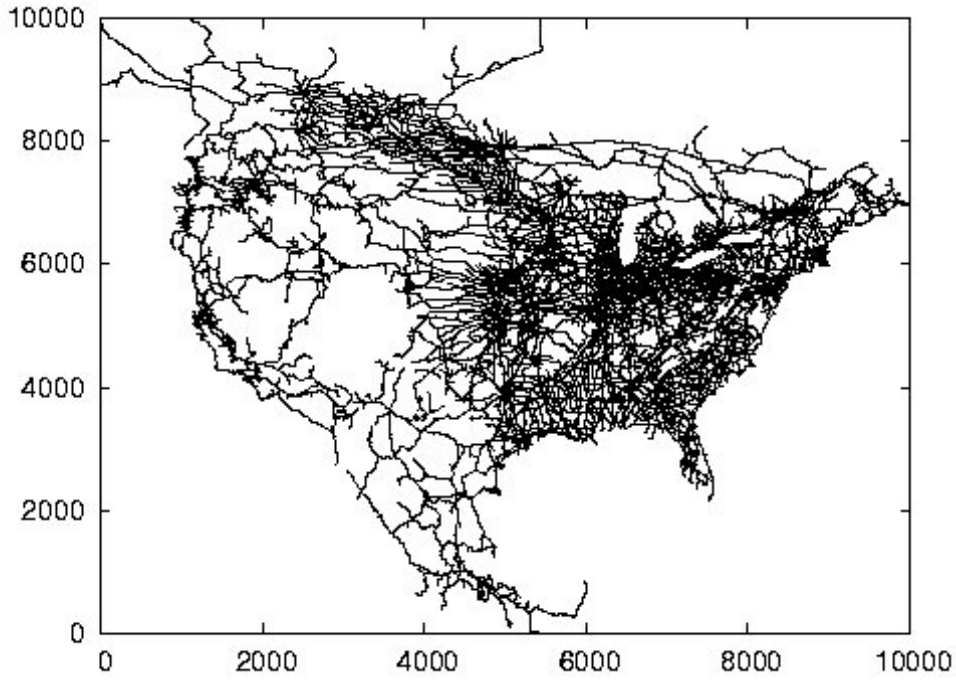

Bild 1: Das Straßennetz von Nordamerika in grafischer Darstellung, erzeugt aus den normalisierten Daten von ***http://www.cs.fsu.edu/~lifeifei/SpatialDataset.htm***

6. Tree-Spanning Aufgabenbeschreibung

- Entwickeln Sie eine Software, mit der ein Netz mit minimalen Kosten berechnet werden kann, genannt Spannender Baum mit minimalen Kosten. Wählen Sie dazu einen Tree-Spanning-Algorithmus aus der Vorlesung, der für diese Aufgabe am besten geeignet ist. Zur Verwaltung der Daten müssen Fibonacci-Heaps als Hilfsdatenstruktur verwendet werden.
- Die Ergebnisse der Netzberechnung sind in geeigneter Form darzustellen. Ebenso soll der Verlauf des Algorithmus anhand einer grafischen Darstellung beobachtbar sein.

- Parallel zum Ablauf des Tree-Spanning-Algorithmus soll ein Monitoring für die Priority-Queue-Struktur durchgeführt werden. Überlegen Sie sich, welche Informationen dazu sinnvoll und notwendig sind und wie diese dargestellt werden.
- Testen Sie die Anwendung zunächst für kleine ungerichtete Graphen. Testen Sie danach mit realen Daten. Verwenden Sie als ungerichteten Graph Straßennetze, die als Dateien unter ***http://www.cs.fsu.edu/~lifeifei/SpatialDataset.htm*** heruntergeladen werden können. Führen Sie für diese Daten auch experimentelle Analysen durch. Jedes Analyse-Experiment ist sorgfältig zu planen, durchzuführen und zu dokumentieren. Was wird analysiert? Welche Daten sind dafür notwendig? Wie wird der Test durchgeführt und protokolliert? Wie werden die Ergebnisse numerisch und grafisch dargestellt?

Eine wichtige Bitte

Lücken im Stoffverständnis erschweren bekanntlich das Verständnis der darauffolgenden Themen. Wenn Sie Fragen oder den Stoff noch nicht richtig verstanden haben, dann kommen Sie bitte rechtzeitig zu mir, am besten persönlich, denn fachliche Fragen sind per Email nicht immer gut zu diskutieren. Falls ich gerade mal keine Zeit habe, können wir Termine zeitnah vereinbaren. Insbesondere nutzen Sie bitte auch die Übungen dazu, um Fragen zu stellen und zu diskutieren. Denken Sie aber auch daran, dass Fragen gut vorbereitet werden müssen. Vertiefen Sie sich also zunächst in den Stoff und kommen dann mit gut vorbereiteten Fragen zu mir.

Literaturempfehlungen

1. **Thomas Cormen**, Charles E. Leiserson, Ronald E. Rivest, Clifford Stein. Introduction to Algorithms. The MIT Press. 2009 3rd edition. ISBN-13: 978-0262533058 (Hardcover empfehlenswert, da Buch sehr dick und schwer).
2. **Thomas Cormen** et al. Algorithmen – Eine Einführung. 2. Auflage, Oldenbourg 2007, ISBN-13: 978-3486275151. Ca. 70 €. (deutsche Fassung der engl. Ausgabe Intro. to Alg.)
3. **Sara Baase**, Allen Van Gelder. Computer Algorithms. 3rd Edition Addison Wesley 2000, ISBN: 978-0201612448
4. **Michael T. Goodrich**, Roberto Tamassia. Algorithm Design. Wiley 2002. ISBN 0471383651 (ca. 50 €)
5. **Th. Ottman**, P. Widmeyer. Algorithmen und Datenstrukturen. 4. Auflage 2002. Spektrum Akademischer Verlag. ISBN-13: 978-3827410290.