Themen für das Praktikum Algorithmen

TECHNISCHE UNIVERSITÄT DARMSTADT

Wintersemester 2024/25

- Bewerbungen auf Praktikumsthemen werden bis zum **28.10.2024** um **08:00** unter der E-Mail Adresse **lab@algo.informatik.tu-darmstadt.de** angenommen. Danach ist keine Bewerbung mehr möglich. Bewerbungen unter anderen E-Mail Adressen werden ignoriert. Jede Bewerbung muss drei Themen mit Prioritäten (Erst-, Zweit- und Drittwunsch) enthalten. Zur Bewerbung auf ein Thema schreiben Sie uns bitte eine kurze Motivation, warum Sie genau dieses Thema bearbeiten möchten sowie Ihre Vorkenntnisse.
- Das Kick-Off Treffen findet nach Vereinbarung für jedes Thema separat statt.
- Um die Integration der Praktikumsergebnisse in das quelloffene Projekt MOTIS zu ermöglichen, müssen alle Teilnehmer Ihre Abgaben unter die MIT / Apache2 Lizenz stellen. Nach dem Kick-Off senden Sie dem Betreuer des Themas eine unterschriebene Lizenzvereinbarung (MIT / Apache2) und falls nötig eine unterschriebene Vertraulichkeitserkärung für die für das Praktikum bereitgestellten Daten. Dies wird gleichzeitig als verbindliche Zusage zur Bearbeitung des Themas angesehen.
- In einem zweiwöchigen Rythmus finden regelmäßige verpflichtende Treffen statt. Die genauen Termine werden beim Kick-Off Treffen vereinbart. Bei jedem Termin stellt jeder Teilnehmer einen Foliensatz mit 3 Folien vor (max. 5 min):
  - Welche Fortschritte / Erkenntnisse wurden seit dem letzten Treffen erreicht?
  - Welche nächsten Schritte sind geplant?
  - Liste mit Fragen - bitte achten Sie darauf, dass die Bewertungskriterien unter anderem selbstständiges Arbeiten berücksichtigen. Hier sollten Sie folglich nur Fragen stellen, deren Beantwortung nicht durch Google / bzw. das Lesen von bereitgestelltem Material möglich ist.
- Die endgültige Abgabe besteht aus:
  - Code, der für dieses Praktikum geschrieben wurde (beinhaltet Code, der Auswertungen, Tests oder ähnliches durchführt).
  - Einen Praktikumsbericht, in dem Sie beschreiben, welche Probleme Sie wie gelöst haben und wieso Sie diesen konkreten Ansatz gewählt haben.

Es gelten folgende Markierungen:

- **A**: Das Thema kann als Praktikum Algorithmen (6CP) oder Vertiefungspraktikum Algorithmen (6CP) bearbeitet werden.
- **P**: Das Thema kann als Projektpraktikum (9CP) bearbeitet werden.
- **E**: Das Thema kann einzeln bearbeitet werden.
- **G**: Das Thema kann auch als Gruppe bearbeitet werden.

Der Umfang des Themas wird den Umständen entsprechen angepasst. Die Praktikumsform 'Projektpraktikum', als auch Gruppenarbeit, erhöhen selbstverständlich den Arbeitsumfang.

Der Abgabetermin des Praktikums ist der 31.03.2025.

Hinweise:

- Planen Sie bitte zusätzlichen Aufwand ein, falls die genannten Programmiersprachen (C++, Typescript, Javascript, CUDA) oder verwendeten Technologien (z.B. Git, CMake, React, etc.) neu für Sie sind.
- Betrifft alle C++ Praktika: Als Referenz für die Komplexität des Projektes, das im Praktikum erweitert wird, können Sie MOTIS[1] anschauen.
  Bitte überlegen Sie sich genau, ob Sie mit einer großen "real-world" Modern-C++ Code Basis klarkommen. Bisherige Erfahrungen haben gezeigt, dass Studierende den Einarbeitungsaufwand und die Komplexität der Programmiersprache unterschätzen.
  Einen Crash-Kurs in die MOTIS-typische C++-Programmierung gibt es hier: `https://www.algo.informatik.tu-darmstadt.de/dl/cpp.pdf`.
- Falls Themen in Gruppen bearbeitet werden, steigt der Aufwand linear mit der Summe der CPs der Gruppenmitglieder. Die unten genannten Themen sind auf Einzelpersonen ausgelegt. Bei einer Bearbeitung in der Gruppe (nur bei Themen mit **G** möglich!) wird der Umfang entsprechend erweitert.

[1] `https://github.com/motis-project/motis`

Thema 1: Erhöhen der Datenlokalität des Fahrplans für den RAPTOR Algorithmus [A, P, E]

Für den Public Transit Router `nigiri`[2] soll untersucht werden, ob durch Permutation der Fahrplandaten eine Verbesserung der Performance des RAPTOR Algorithmus erreicht werden kann.

Dazu sind folgende Schritte notwendig:

- Implementieren einer Funktion zur Umsortierung der Fahrplandaten anhand einer Permutation als Eingabe
- Generation von Permutationskandidaten zur Verbesserung der Lokalität
- Evaluation der Performance der Kandidaten

Programmiersprache: C++

Thema 2: Berechnung von Origin-Destination Matrizen in MOTIS [A, P, E, G]

Zur Untersuchung der Verkehrsanbindung kommen Origin-Destination (OD) Matrizen zum Einsatz. Eine OD Matrix kodiert in jeder Zeile für eine Quelle q die Eigenschaften $e(q, s)$ für alle Senken s. Die Eigenschaft $e(q, s)$ kann z.B. die Reisezeit zwischen der Quelle q und der Senke s sein.

Da Reisen mit dem öffentlichen Verkehr inhärent zeitabhängig sind, also die Reisezeit von der gewählten Startzeit abhängig ist, ist auch die OD Matrix für diese Eigenschaft zeitabhängig. Soll ein längerer Zeitraum betrachtet werden, so kann eine dreidimensionale Matrix aus den zeitabhängigen OD Matrizen erzeugt werden.

Zur räumlichen Analyse der Verkehrsanbindung kann der zu untersuchende Raum in ein Raster (z.B. aus Zellen der Größe 100 m x 100 m) eingeteilt werden. Von den Mittelpunkten der Zellen sollen mit dem intermodalen Routing System `MOTIS`[3] die Reisezeiten zu allen anderen Zellmittelpunkten ermittelt werden.

Hierbei können die ersten und letzten Meilen für jede Zelle vorberechnet werden und dann mittels dem RAPTOR Algorithmus ein One-to-all Routing durchgeführt werden. Für jede Abfahrt/Ankunft des Fahrplans soll ein Routing durchgeführt werden. Wobei ein Routing auch mehrere Abfahrten/Ankünfte abdecken kann, falls diese vom Zellmittelpunkt aus zur gleichen Zeit stattfinden.

Im Falle eines Projektpraktikums oder eine Gruppenarbeit ist zusätzlich eine (interaktive) visuelle Darstellung der erzeugten Daten zu implementieren.

[2] `https://github.com/motis-project/nigiri`
[3] https://github.com/motis-project/motis

Programmiersprache: C++, TypeScript / Svelte / Maplibre / deck.gl (UI)

Thema 3: One-to-many RAPTOR Visualisierung [A, E]

Bei einem one-to-many Routing werden ausgehend von einem Startpunkt die Reisezeiten zu mehreren Zielpunkten berechnet. Wird dies in Form einer Rückwärtssuche durchgeführt, werden für ein Ziel die Reisezeiten ausgehend von mehreren Startpunkten ermittelt. Mit diesem Ansatz lassen sich interessante Fragestellungen beantworten, z.B.:

- Ich plane eine Anreise am Vortag in einer fremden Stadt an. Welche Hotels liegen günstig, wenn ich am nächsten Tag um 9 Uhr einen Termin an einem bestimmten Ort habe?
- Gegeben sei eine Menge von Geschäften. Was sind die Einzugsbereiche der Geschäfte? Gibt es Gebiete aus denen man schlecht ein Geschäft erreicht?

Der Public Transit Router `nigiri`[4] unterstützt bereits ein one-to-one Routing und verwendet dafür den RAPTOR Algorithmus, der von seiner grundlegenden Funktionsweise ein one-to-all Routing durchführt und daher leicht dafür angepasst werden kann. Um diese Funktionalität nutzbar zu machen, soll eine Benutzeroberfläche implementiert werden, mit der das one-to-many Routing verwendet werden kann. Ziel ist es Fragestellungen, wie die oben genannten, zu beantworten, und eine Visualisierung der Ergebnisse anzuzeigen. Daher ist auch die Definition einer geeigneten API für die Kommunikation zwischen Front- und Backend notwendig.

Programmiersprache: TypeScript / Svelte / Maplibre / deck.gl (Frontend), C++ (Backend)

[4] `https://github.com/motis-project/nigiri`