Technische Universität München
Institut für Informatik
Prof. Dr. Angelika Steger

SS 1999
Aufgabenblatt 1
11. Mai 1999

Praktikum Diskrete Optimierung

(Abgabetermin: Dienstag, den 18. Mai 1999, 12.00 Uhr)

Aufgabe 1 Einführung in LEDA

Machen Sie sich mit der Benutzung der LEDA-Bibliothek bei der Erstellung von C++–Programmen vertraut. Legen Sie auf der sunhalle ein Verzeichnis an, in dem Sie im Verlauf des Semesters Ihre Praktikumsaufgaben bearbeiten und Ihre Lösungen erstellen werden, und holen Sie sich das Makefile und das Sourcefile ledademo.cc von der Praktikumswebpage in dieses Verzeichnis. Das Programm ledademo sollte sich nun mittels `make ledademo` übersetzen lassen.

Aufgabe 2 Grundlegende Datenstrukturen

Erstellen Sie eine Reihe von grundlegenden Datenstrukturen zur Modellierung des Problems. Da Sie diese Klassen/Strukturen/Typen im gesamten Praktikum wiederverwenden werden, sollten Sie hierbei auf eine effiziente und modulare Implementierung achten. Beispielsweise könnten Sie folgende Teile des Modells als getrennte Typen realisieren:

> Pfade, Kanten, ein gesamtes Netzwerk, eine mögliche Lösung, die Nachbarschaft eines Knotenpaares.

Für diese Klassen[1] sollten folgende Funktionen realisiert werden:

> Einlesen von Problemen, Einlesen und Abspeichern von Lösungen, Berechnung des Werts der Zielfunktion zu einer Lösung, Erzeugung aller Nachbarn eines Paares, Graphische Anzeige einer Lösung, Erzeugung einer zufälligen Lösung.

Für diese Datenstrukturen sollten Sie ein eigenes .cc- und .h-File anzulegen, damit Sie diese Funktionalität bei späteren Aufgaben komfortabel nutzen können.

Aufgabe 3 Bewertung von Lösungen

Schreiben Sie ein Programm

```
evalSol <basename>
```

[1] Im Folgenden werden wir einzelne Typen als Klassen bezeichnen. Allerdings müssen Sie die objektorientierten Features von C++ nicht verwenden, sondern können stattdessen auch struct's und eigenständige Funktionen verwenden.

welches das Problem basename.def und eine zugehörige Lösung basename.sol einliest, diese graphisch darstellt (mit LEDA-GraphWin) und den Wert der Zielfunktion berechnet. Hierbei können Sie vom Programmrahmen `ledademo.cc` ausgehen. Bei der graphischen Darstellung sollte der Typ der verwendeten Kanten farblich und/oder durch die Linienbreite visualisiert werden.

Aufgabe 4 Best-First-Suche

Realisieren Sie ein Programm

```
bestfirst <max. execution time in min.> <basename>
```

das ausgehend von einer zufälligen Lösung eine Best-First-Suche durchführt. Die als Kommandozeilenargument angegebene Laufzeit soll von Ihrem Programm nicht überschritten werden (ggf. muß das Programm also vorzeitig stoppen).

Hinweise

Die Abgabe der Lösungen muß jeweils bis Dienstag 12.00 Uhr per E-Mail an die Adresse `algoprak@hpmayr1.informatik.tu-muenchen.de` erfolgen. Es sind nur die C++–Sourcen der erstellten Programme abzugeben, also keine übersetzten Executables.
Die Aufgabenblätter und Eingabedaten, sowie alle anderen Daten, die zur Bearbeitung der Aufgaben nötig sind, können im WWW über die folgende URL bezogen werden:

`http://wwwmayr.in.tum.de/lehre/1999WS/optprak/data.html`

Anforderungen an abgegebene Programme

Die abgegebenen Lösungen werden nach den folgenden Kriterien bewertet:

- Korrektheit der berechneten Ergebnisse
- Effizienz der Implementierung
- Lesbarkeit des Quelltexts (ausreichend Kommentare, die das Verständnis erleichtern)

Außerdem legen wir großen Wert darauf, daß die Lösungen selbst erarbeitet und nicht Programme anderer Teilnehmer als "Vorlage" verwendet werden. Diskussionen über Probleme bei der Implementierung und Austausch von Ideen sind selbstverständlich erlaubt, lediglich die direkte Weitergabe von Source-Code ist nicht gestattet. Sollten Sie bei gewissen Problemen trotz ernsthafter Bemühungen nicht weiterkommen, stehen Ihnen die Praktikumsbetreuer gerne für Fragen zur Verfügung.

Praktikumsbetreuer

- Alexander Hall, Raum S2214, Tel. 289-25342, hall@in.tum.de
- Thomas Schickinger, Raum S2233, Tel. 289-25781, schickin@in.tum.de