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

SS 1999
Merkblatt 2
11. Mai 1999

Praktikum Diskrete Optimierung
Hinweise zur Benutzung von LEDA

1 Allgemeines

LEDA (Library of Efficient Datatypes and Algorithms) ist eine Bibliothek von C++-Klassen, die am MPI in Saarbrücken entwickelt wurde und die eine Vielzahl von höheren Datenstrukturen sowie Hilfsmittel zur Visualisierung und Animation zur Verfügung stellt. Für Zwecke der Forschung und der Ausbildung darf LEDA frei benutzt werden; für kommerzielle Anwendungen muß eine Lizenz käuflich erworben werden. Näheres zu LEDA ist im WWW unter der folgenden URL zu finden:
`http://www.mpi-sb.mpg.de/LEDA`
Dort können vorübersetzte Bibliotheken für eine ganze Reihe von Plattformen heruntergeladen werden. Im Praktikum werden wir LEDA 3.7.1 benutzen.

2 Installation

LEDA ist unter `/usr/wiss/schickin/local/dist/LEDA` auf den sunhalle-Rechnern installiert. In diesem Verzeichnis liegt auch das komplette Manual als mit gzip komprimiertes PostScript-File `MANUAL.ps.gz`. Um mit LEDA zu arbeiten, sind folgende Environment-Variablen zu setzen:

```
export LEDAROOT=/usr/wiss/schickin/local/dist/LEDA
export LD_LIBRARY_PATH=$LEDAROOT:$LD_LIBRARY_PATH
```

Mit `$LEDAROOT/Manual/cmd/xlman` kann der Online-Manual-Viewer von LEDA gestartet werden. Alternativ kann das Online-Manual auch im WWW unter `http://www.mpi-sb.mpg.de/LEDA/MANUAL/MANUAL.html` ansehen.
LEDA kann ohne Probleme auch auf einem PC unter Linux oder unter Windows (z.B. für Visual C++) installiert werden. Dazu kann man sich die passende Version über obige URL bei der LEDA Software GmbH besorgen (kostenlose *research license*).

3 Verwendung

Um ein C++-Programm, das LEDA verwendet, zu übersetzen und zu linken, müssen dem Compiler das Verzeichnis mit den LEDA-Includes und die LEDA-

Bibliotheken bekannt gemacht werden. Ein entsprechendes `Makefile` sowie ein Beispielprogramm `ledademo.cc` liegen auf den Praktikumsseiten im WWW als Vorlage bereit. Nach Kopieren der Dateien in ein eigenes Verzeichnis sollte sich `ledademo.cc` mittels `make ledademo` übersetzen lassen. Ein selbst erstelltes Programm `foo.cc` kann dann mittels `make foo` übersetzt werden.
Soll eine Klasse `xyz` aus LEDA im eigenen C++-Programm verwendet werden, so muß nur mittels `#include <LEDA/xyz.h>` der entsprechende Header eingebunden werden. Wir empfehlen, daß Sie sich mit folgenden Klassen vertraut machen:

> `array/array2` (etwas komfortabler als herkömmlich C++-Arrays), `matrix/vector` (lineare Algebra zum Hausgebrauch), `string` (ähnlich `char *` in C++, aber mit zusätzlichen Features), `random_source` (Erzeugung von Zufallszahlen), `stack`, `queue`, `list`, `set` (Menge von Elementen), `map` (Abbildung von einem Typ auf einen anderen),
> `GraphWin` (Darstellung von Graphen auf dem Bildschirm und Interaktion mit dem Benutzer).

Eventuell könnten noch die folgenden Klassen von Interesse sein:

> `partition` (Partition einer Menge),
> `p_queue` (Prioritäts-Warteschlange), `graph` (LEDA-Graph),
> `node/edge_array` (Zuordnung von Werten zu Knoten bzw. Kanten),
> `node/edge_map` (dynamische Variante von `node/edge_array`),
> `node/edge_set` (Menge von Knoten bzw. Kanten),
> `node_partition` (Partition der Knotenmenge eines Graphen),
> `node_pq` (Prioritäts-Warteschlange von Knoten eines Graphen),
> `color` (Definition von Farben), `window` (Bildschirm-Fenster)

Näheres zur Funktionalität und Benutzung dieser Klassen sollte z.B. im Online-Manual-Viewer nachgelesen werden. Außerdem stehen in `LEDA/basic.h` verschiedene nützliche Funktionen zur Verfügung, die im Manual-Abschnitt "misc" erklärt sind.
Als einfaches Beispiel sei kurz die Benutzung einer Liste aus LEDA skizziert. Nach `#include <LEDA/list.h>` steht der Template-Typ `list<T>` zur Verfügung, wobei `T` einen beliebigen Typ für die Elemente der Queue spezifizieren kann. Etwa kann mit `list<string> Q` eine Liste von Strings deklariert werden. An diese Liste kann mit `Q.append(v)` ein Knoten angehängt werden. Mit `forall (s, Q)` kann über alle Strings in der Liste iteriert werden.