RWTH Aachen
Lehrgebiet Theoretische Informatik
Kneis—Langer—Rossmanith

WS 2009
Aufgabenblatt 2
9. November 2009

Softwareprojekt-Praktikum Graphalgorithmen

Aufgabe 7

Falls noch nicht vorhanden, erstellen Sie eine praktische Arbeitsumgebung im SVN mit Unterverzeichnissen für die Graphbibliothek (z.B. `graphbib/`), Dokumentation (z.B. `doc/`), Tests (z.B. `tests/`), sowie einfachen Beispielen zur Anwendung (z.B. `examples/`).

Nach Möglichkeit soll zu jeder Funktion der Graphbibliothek ein kleines Anwendungsbeispiel vorhanden sein. Erweitern Sie diese Sammlung im Laufe des Praktikums sinnvoll.

Außerdem soll es (uns) möglich sein, `make tests` aufzurufen, so daß die Graphbibliothek vollständig übersetzt und alle Tests und Unit-Tests aufgerufen und überprüft werden. Erstellen Sie also entsprechende Makefiles.

Aufgabe 8

Entwerfen (und dokumentieren) Sie Algorithmen für Tiefen- und Breitensuche auf gerichteten und ungerichteten Graphen: Welche Anforderungen gibt es? Wie kann ein Anwender auf Ereignisse reagieren, z.B. erster oder letzter Besuch eines Knotens? Welche Übergabeparameter sind aufgrund dieser Anforderungen sinnvoll, welche Rückgabewerte?

Erweitern Sie die Spezifikation Ihrer Graphbibliotek um die entsprechenden Funktionen inklusive vollständiger Signatur, *bevor* Sie die Algorithmen implementieren.

Entwerfen und implementieren Sie Test-Cases für die Algorithmen, *bevor* Sie die Algorithmen selbst implementieren. Dieses ist glücklicherweise möglich, da die Signatur bereits in der Spezifikation angegeben ist.

Implementieren Sie nun die Algorithmen möglichst effizient.

Sinnvollerweise werden die Aufgaben für Test-Cases und Implementierung der Algorithmen von verschiedenen Personen übernommen.

Welche Laufzeit haben Ihre Algorithmen?

Aufgabe 9

Entwerfen und implementieren Sie einen Algorithmus, der testet, ob ein gerichteter oder ungerichteter Graph azyklisch ist. In dieser Aufgabe soll zusätzlich die Idee der zertifizierenden Algorithmen aufgegriffen werden: Auf Wunsch erhält der Anwender einen Beweis dafür, daß der Graph azyklisch ist oder nicht. Wie kann ein solcher Beweis aussehen?

Selbstverständlich soll analog zu Aufgabe 8 vorgegangen werden: Erst Spezifikation und Dokumentation, dann Test-Cases und Implementierung von verschiedenen Personen!

Aufgabe 10

Entwerfen und implementieren Sie Algorithmen, die die Zusammenhangskomponenten ungerichteter Graphen.

Jeweils eine Version soll die Komponenten als Knotenmengen zurückgeben, und eine Version als die von diesen Komponenten induzierten Untergraphen.