Dr. Matthias Grabert
Claudia Fischer

Abteilung Angewandte Informationsverarbeitung

7. Juli 2004
Blatt 9

C++ mit Data Mining Anwendungen (SS 2004)

Abgabetermin: 22. Juli 2004

Die Standard Template Library

Erdös- Zahlen* (2 Punkte)

Paul Erdös (1913 – 1996) war ein berühmter ungarischer Mathematiker, der Zeit seines Lebens eine ganz Menge publiziert hat. Weil es eine Ehre ist, gemeinsam mit so einem berümten Mann zu publizieren, wollten natürlich möglichst viele Leute mit ihm zusammen veröffentlichen. Leute, die nicht direkt mit ihm publiziert haben, sind auch schon stolz, wenn es einen Weg zwischen ihnen und Erdös im „Publikationsgraphen“ gibt. (Dieser enthält für jeden Menschen einen Knoten und zwischen zwei Knoten gibt es genau dann eine Kante, wenn diese beiden Personen miteinander publiziert haben.) Weil Erdös sich auch viel mit Graphentheorie beschäftigt hat, kam die Idee der Erdös-Zahlen auf. Die Erdös-Zahl einer Person gibt die Länge des kürzesten Weges zwischen Erdös und ihr im Publikationsgraph an. Gibt es keinen solchen Weg, so ist sie unendlich.
Die Datei erdoes.txt enthält zu jeder Person mit Erdös-Zahl 1 (diese stehen jeweils linksbündig in der Datei) die Personen mit denen sie publiziert hat (diese sind um einen Tabulator eingerückt in der Datei) – außer Erdös. Die Personen mit Erdös-Zahl 1 sind mit Großbuchstaben geschrieben. Bei den anderen handelt es sich also um „Erdös-2-Leute“. Bestimmen sie zu allen „Erdös-2-Leuten“ jeweils die Anzahl von „Erdös-1-Leuten“, mit denen sie publiziert hat und geben sie die Personen mit Erdös-Zahl 2 alphabetisch sortiert aus. Geben Sie außerdem zu jeder dieser Personen die Anzahl der Personen mit Erdös-Zahl 1 aus, mit denen sie schon eine Veröffentlichung hatte. Die Datei erdoes.txt soll ihrem Programm über die Standardeingabe zugeführt werden.
Zur Lösung dieser Aufgabe sollen Sie ein assoziatives Array verwenden.

Shunting Yard- Algorithmus* (4 Punkte)

Schreiben Sie ein kurzes Programm, dass einen arithmetischen Ausdruck von Infix- Notation in Postfix- Notation transformiert. (Unter Infix- Notation versteht man die Notation, die man üblicherweise gebraucht.)
Benützen Sie den Shunting Yard- Algorithmus um dieses Problem zu lösen. Hier die Regeln: Sie benötigen als Hilfsmittel einen Stack. Betrachten Sie nun den Infix- Ausdruck zeichenweise (von links nach rechts).

- Ist das Zeichen ein Operand, so kommt es sofort in den Ausgabestring.
- Handelt es sich bei dem Zeichen um eine öffnende Klammer, so kommt diese auf den Stack.
- Wenn das Zeichen eine schließende Klammer ist, dann werden alle Operatoren vom Stack entfernt und ausgegeben, bis zur öffnenden Klammer. Beide Klammern werden dann gelöscht.
- Falls das Zeichen ein Operator ist, werden alle Operatoren auf dem Stack entfernt und ausgegeben, bis das oberste Stack- Element von geringerer Priorität ist oder der Stack leer ist (steigende Priorität: $+-, */$). Dann wird der Operator auf den Stack gelegt.
- Am Ende müssen dann noch alle Operatoren auf dem Stack ausgegeben werden.

Ein Beispiel:

- Infix- Notation: (A + B)*(C - D)
- Präfix- Notation: AB+ CD-*

Der Infix- Ausdruck soll beim Aufruf des Programms übergeben werden. Dabei sollen die einzelnen Zeichen durch Leerzeichen getrennt sein.

Problem des Handlungsreisenden* (4 Punkte)

Im folgenden sollen Sie ein Programm schreiben, das möglichst einfach das Problem des Handlungsreisenden löst. Ein Handlungsreisender hat eine Reihe von Städten zu besuchen und sucht die Rundreise, die durch jede Stadt führt und dabei möglichst kurz ist. Dabei reist der Handlungsreisende mit einem Flugzeug, so dass sich die Länge des Wegs aus der Entfernung der Punkte ergibt. Das Problem ist berühmt, weil es eine auerordentliche praktische Bedeutung hat, aber nur mit hohem rechnerischem Aufwand gelöst werden kann. Sie können davon ausgehen, dass der Handlungsreisende genau vier Städte besucht, deren Koordinaten über die Standarteingabe eingelesen werden sollen. Ihr Programm soll dann den kürzesten Weg berechnen und ausgeben.
Implementieren Sie dies mithilfe einer Liste.

Viel Erfolg!