

Übungsblatt 10/11

ALGEBRAISCHE GRUNDLAGEN und FORMALE SPRACHEN

Abgabe: bis Freitag 21.01.2005, 13:00 Uhr in die Einwurfkästen im Untergeschoß des neuen Informatikgebäudes am Fasanengarten

**Erreichbare Punkte: 15 P / 75 T
(Praktische Aufgaben / Theoretische Aufgaben)**

Hinweis:

Alle Programmieraufgaben sind unter Verwendung der in der Vorlesung vorgestellten In-/Out-Klassen zu erstellen und (ähnlich wie die Beispielprogramme im Skript) in ausführlicher Form zu kommentieren.

Versehen Sie die Programme mit Eingabeüberprüfungen, die sicherstellen, dass der Anwender korrekte Daten eingegeben hat.

1 VORANGEGANGENE KURSEINHEITEN

1.1 Lotto mit Sortierung (5P)

In diese Aufgabe soll das Programm aus Aufgabe 3.1 auf Übungsblatt 8 um eine Methode zur Sortierung der gezogenen Zahlen erweitert werden. Dabei soll als Sortierverfahren Sortieren durch Auswählen verwendet werden. Das Verfahren funktioniert folgendermaßen:

1. Suche im Feld das kleinste Element und vertausche es mit dem Element an der ersten Feldposition.
2. Jetzt wird die Sortierung auf den Teil des Feldes ab dem zweiten Element beschränkt: Suche hier das kleinste Element und vertausche es mit dem an Position zwei.
3. Führe dieses Vorgehen für den Rest des Feldes fort: Vor dem i -ten Schritt ist das Feld bis zur Position $i-1$ aufsteigend sortiert. Suche im Bereich ab i das kleinste Element und vertausche es mit dem i -ten Element.

Schreiben Sie die Methode `void sort(int[] numbers)` für ihre Klasse `Lotto`, die genau das Obige leistet. Dieser wird ein Array ganzer Zahlen übergeben. Nach der Ausführung soll das Feld mittels Sortieren durch Auswählen sortiert worden sein.

Erweitern Sie zudem das Programm `Lotto` so, dass die Zahlen sortiert ausgegeben werden.

2 RELATIONEN

2.1 Eckenanzahl und Kantenanzahl bei Graphen (10T)

Wie berechnet sich die Kantenanzahl aus der Eckenanzahl

- a) für einen vollständigen, gerichteten Graphen mit reflexiven Kanten? (5T)
- b) für einen vollständigen, ungerichteten Graphen mit reflexiven Kanten? (5T)

Formulieren Sie Ihre Annahme und beweisen Sie diese mit Hilfe der vollständigen Induktion.

2.2 Reflexive transitive Hülle (10P)

Erstellen Sie ein Java-Programm, welches die reflexive transitive Hülle einer Adjazenzmatrix eines Graphen G mit Hilfe des aus der Vorlesung bekannten Warshall-Algorithmus berechnet. Die Graphen sollen 10 Knoten (0 bis 9) enthalten.

Die Eingabe eines Graphen soll folgendermaßen erfolgen:

Der Anwender gibt zwei Knoten als Start- und Endknoten einer Kante im Graphen ein. Wenn diese Kante noch nicht im Graphen enthalten ist, wird sie eingefügt. Wenn sie bereits enthalten war, wird sie wieder entfernt. In beiden Fällen wird anschließend der bisher eingegebene Graph als Adjazenzmatrix ausgegeben und dem Anwender die Möglichkeit geboten, ein weiteres Knotenpaar einzugeben.

Wenn der Anwender bei Eingabe eines Knotens eine Zahl größer als 9 eingibt, bricht die Knoteneingabe ab und die Berechnung und Ausgabe der reflexiven transitiven Hülle des Graphen beginnt.

Beispiel:

Geben Sie die Kanten durch den Start- und Endknoten ein (Eine Zahl > 9 beendet die Eingabe der Knoten).

{hier erfolgt die Eingabe der Kanten}

Geben Sie die Nummer des ersten Knotens ein:

0

Geben Sie die Nummer des zweiten Knotens ein:

8

0 1 0 0 0 0 0 0 1 0

0 0 1 0 0 0 0 0 0 0

0 0 0 1 0 0 0 0 0 0

1 0 0 0 0 0 0 0 0 0

0 0 0 0 0 1 0 0 0 0

0 0 0 0 0 0 1 0 0 0

0 0 0 0 0 0 0 1 0 0

0 0 0 0 1 0 0 0 0 0

0 0 0 0 0 0 0 0 0 1

1 0 0 0 0 0 0 0 0 0

Geben Sie die Nummer des ersten Knotens ein:

10

Eingegebene Adjazenzmatrix von G :

0 1 0 0 0 0 0 0 1 0

0 0 1 0 0 0 0 0 0 0

0 0 0 1 0 0 0 0 0 0

1 0 0 0 0 0 0 0 0 0

0 0 0 0 0 1 0 0 0 0

0 0 0 0 0 0 1 0 0 0

0 0 0 0 0 0 0 1 0 0

0 0 0 0 1 0 0 0 0 0

0 0 0 0 0 0 0 0 0 1

1 0 0 0 0 0 0 0 0 0

Reflexive transitive Hülle von G :

1 1 1 1 0 0 0 0 1 1

1 1 1 1 0 0 0 0 1 1

1 1 1 1 0 0 0 0 1 1

1 1 1 1 0 0 0 0 1 1

0 0 0 0 1 1 1 1 0 0

0 0 0 0 1 1 1 1 0 0

0 0 0 0 1 1 1 1 0 0

0 0 0 0 1 1 1 1 0 0

1 1 1 1 0 0 0 0 1 1

1 1 1 1 0 0 0 0 1 1

4.2 Markov: Wortlänge halbieren (6T)

Geben Sie einen Markov-Algorithmus an, der alle Wörter der Sprache $L_1 = \{ \varepsilon, l^n \mid n \in \mathbb{N} \}$ in der Länge (abgerundet) halbiert. Verwenden Sie nicht mehr als 5 Regeln.

4.3 Markov: Strich / Klammer-Umwandlung (11T)

In dieser Aufgabe sollen zwei gesteuerte Markov-Algorithmen angegeben werden, die eine ganze Zahl n (dargestellt durch eine Strichfolge) durch einen Klammersausdruck der Form

$$\underbrace{((\dots((\))\dots))}_{2n}$$

ersetzen, d.h. n öffnende Klammerzeichen gefolgt von n schließenden Klammerzeichen.

Beispiel: Das Eingabewort $|||$ wird ersetzt durch die Zeichenreihe $((\))$.

Geben Sie je einen Markov-Algorithmus an, der

- ohne Schiffchen auskommt. (4T)
- mit Schiffchen realisiert wird. (7T)

4.4 Formale Systeme (3T)

- Was beschreiben formale Systeme? (1T)
- Geben Sie zwei Beispiele für formale Systeme. (2T)