Informatik I WS 2005/2006

Übungsblatt 11

Abgabe: 30.1.06 - 1.2.06

Aufgabe 1 (Stacks, Queues und OOD) Punkte: 3+2

a) In der Vorlesung haben Sie die beiden Datentypen `Stack` und `Queue` kennengelernt. Beide Datenstrukturen wurden intern mit Hilfe von Listen implementiert.

Implementieren Sie nun eine Klasse `SQueue`, die intern keine Listen, sondern Stacks verwendet. Für den Benutzer darf es keinen Unterschied zwischen, `SQueue` und `Queue` geben, das heißt, dass alle für `Queues` definierten Methoden auch für `SQueue` existieren müssen.

Tip: Verwenden Sie in der `dequeue`-Methode einen temporären Stack für die Zwischenspeicherung der Elemente.

b) Entwerfen Sie einen Unit-Test zum Testen der Funktionalität der Klasse.

Aufgabe 2 (Multi-Listen) Punkte: 7+5

a) Implementieren Sie eine Klasse SPMatrix zur Repräsentation quadratischer, dünn besetzter Matrizen. Verwenden Sie dazu die aus der Vorlesung bekannten Multi-Listen. Die Klasse soll mindestens folgende Funktionen haben:

- Dem Kontruktor soll die Anzahl der Spalten und Zeilen übergeben werden.
- Implementieren Sie eine Methode `set( zeile, spalte, wert )` zum Setzen der Matrixeinträge.
- Implementieren Sie eine Methode `get( zeile, spalte )` die den Wert eines Matrixeintrags zurückliefert.
- Überladen Sie den `+-Operator`. Mittels + sollen 2 Matrizen addiert werden.
- Überladen Sie den `*-Operator`. Mittels * sollen 2 Matrizen multipliziert werden.
- Entwerfen Sie eine Funktion `transpose`, die die transponierte Matrix zurück liefert (d.h. die Einträge der Spalten und Zeilen werden vertauscht) .
- Implementieren Sie eine Funktion `print` zur Ausgabe der Matrix.

b) Man bezeichnet $P \in \mathbb{R}^{n \times n}$ als Permutationmatrix, wenn genau ein Elemente in jeder Zeile und Spalte den Wert 1 und alle anderen den Wert 0 haben.

Schreiben Sie eine Funktion, die eine zufällige Permutationsmatrix erzeugt. Verwenden Sie dazu die SPMatrix-Klasse aus Aufgabenteil a).

Was liefert `P*P.transpose()` wenn P eine Permutationsmatrix ist?

Was ergibt P^n bzw P^{n+1} (also P (n-1) bzw n mal mit sich selbst multipliziert)?

Hinweis: Für eine beliebige Matrix $A \in \mathbb{R}^{n \times n}$ führt die Multiplikation $P * A$ zu einer Permutation der Zeilen und $A * P$ zu einer Permutation der Spalten von A. Die Determinante von Permutationsmatrizen ist immer +/-1. Sie sind nicht kommutativ bezüglich der Multiplikation.

Dünn besetzte Matrizen spielen in sehr vielen Anwendungen eine wichtige Rolle: Beispielsweise kann man Zugverbindungen oder Straßenkarten als dünn besetzte Matrizen speichern. Dazu erstellt man eine Matrix in der für jeden Ort eine Spalte und eine Zeile angelegt wird. Dann werden die Werte der Spalten und Zeileneinträge zu den Nachbarorten auf die Länge der Straßen zwischen den Orten gesetzt.

Mit Hilfe solcher Straßenmatrizen lassen sich dann unter Verwendung spezieller Algorithmen die kürzesten Wege zwischen zwei beliebigen Orten berechnen. Dazu aber mehr im nächsten Semester.

Aufgabe 3 (ADT Stack) Punkte: 3

Welche der folgenden Ausdrücke sind syntaktisch zulässig für den ADT Stack? Welchen Wert haben die korrekten Ausdrücke? Geben Sie bei den nicht zulässigen Ausdrücken eine Begründung an, warum sie nicht korrekt sind.

- top(pop(push(e,push(e,create()))))
- top(push(e,isempty(create())))
- isempty(push(h,pop(push(g,push(f,push(e,create()))))))
- isempty(top(e,push(e,create())))
- top(pop(push(e,top(create()))))
- isempty(top(h,pop(push(e,push(e,create())))))