Übungen zu WS 95/96

Algorithmen und Programmierung 3

- Blatt 11 -

H. Schweppe / M. Walter

Abgabe: Mo. 29.1.96 bis 12.00

Aufgabe 1 (10 P)

Gegeben ein ungerichteter Graph G = (V,E). Zeigen Sie:

a) G zusammenhängend => | E | >= | V | -1

b) $\sum_{v \in V} d(v) = 2 * |E|$, also: Die Summe des Knotengrads aller Knoten ist doppelt so groß wie die Kantenanzahl.

c) Sei | V | = n . Wie groß ist | E | maximal? (Beweis)

d) Gesucht ist ein ungerichteter, azyklischer Graph, der kein Baum ist.

Aufgabe 2 (10 P)

Ein Labyrinth ist eine Matrix, die aus den Zeichen `blank` oder `X` besteht. Es gibt an den Außenseiten genau zwei `blank` - Felder, die Eingang und Ausgang darstellen. Ein Quadrat bestehend aus `blank` ist nicht erlaubt, z.B.

```
XX X
X  X
X  X
XX X
```

Dies ist ein korrektes Labyrinth:

```
XX XXXX
X  XX X
XX XX X
XX    X
X  XXXX
XX XX X
XX XX X
      X
XXXXXXX
```

Sie sollen ein beliebiges Labyrinth durch einen ungerichteten Graphen modellieren. Hinweis: Eingang und Ausgang sollen Knoten des Graphen sein. Entwickeln Sie einen Algorithmus, der ein - im Rahmen einer vernünftigen Größenbeschränkung - beliebiges Lybyrinth in einen Graphen überführt.

Aufgabe 3 (10 P)

Implementieren Sie eine Klasse Graph mit Hilfe einer Adjazenzliste. Zu implementieren sind die Operationen:

```
insertNode: graph x node  -> graph
insertEdge: graph x node x node -> graph
findNeighbors: graph x node -> list[node]
printGraph: graph     (* sehr einfache Ausgabe der Knoten und Kanten *)
```

Verwenden Sie in früheren Übungen entwickelte Programme (Listenklasse!). Sehen Sie die Möglichkeit vor, Knoten zu färben.